09/486805

PCT/JP99/03710

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

08.07.99

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

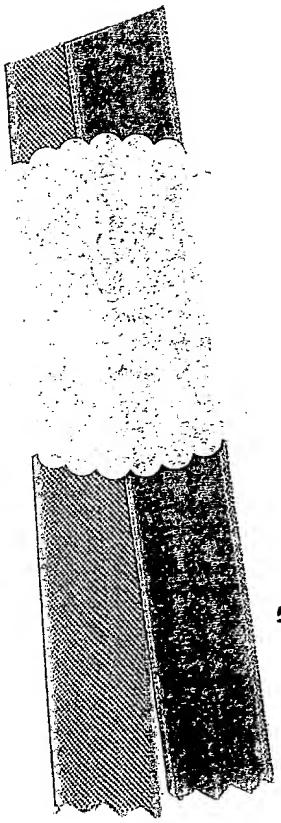
This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: １９９９年 ７月 ２日

出 願 番 号
Application Number: 平成１１年特許願第１８８６６１号

出 願 人
Applicant (s): ソニー株式会社

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)



１９９９年 ７月２９日

特許庁長官
Commissioner,
Patent Office 伊佐山建志

出証番号 出証特平１１－３０５３２７４

【書類名】　特許願

【整理番号】　9900497402

【提出日】　平成11年 7月 2日

【あて先】　特許庁長官殿

【国際特許分類】　G06F　9/45

【発明者】

【住所又は居所】　東京都品川区北品川６丁目７番３５号　ソニー株式会社内

【氏名】　福澤　祐二

【発明者】

【住所又は居所】　東京都品川区北品川６丁目７番３５号　ソニー株式会社内

【氏名】　岡田　徹也

【特許出願人】

【識別番号】　000002185

【氏名又は名称】　ソニー株式会社

【代表者】　出井　伸之

【代理人】

【識別番号】　100091546

【弁理士】

【氏名又は名称】　佐藤　正美

【電話番号】　03-5386-1775

【先の出願に基づく優先権主張】

【出願番号】　平成10年特許願第195586号

【出願日】　平成10年 7月10日

【手数料の表示】

【予納台帳番号】　048851

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9710846

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　コンパイル処理装置およびコンパイル処理方法

【特許請求の範囲】

【請求項１】

ソースプログラムから、そのソースプログラムの解析情報を生成する解析情報生成部と、

前記解析情報に基づいて、第１の実行形式プログラムを生成する第１の実行形式プログラム生成部と、

前記第１の実行形式プログラムに基づいて、プロファイル情報を生成するプロファイル情報生成部と、

前記前記解析情報と、前記プロファイル情報とに基づいて、第２の実行形式プログラムを生成する第２の実行形式プログラム生成部と

を備えることを特徴とするコンパイル処理装置。

【請求項２】

前記解析情報生成部により生成される前記解析情報を記憶する解析情報記憶部を備えることを特徴とする請求項１に記載のコンパイル処理装置。

【請求項３】

前記プロファイル情報生成部により生成される前記プロファイル情報を記憶するプロファイル情報記憶部を備えることを特徴とする請求項１に記載のコンパイル処理装置。

【請求項４】

前記第１の実行形式プログラム生成部と、前記第２の実行形式プログラム生成部とは、１つの実行形式プログラム生成部により構成されることを特徴とする請求項１に記載のコンパイル処理装置。

【請求項５】

前記第１の実行形式プログラム生成部は、目的とする計算機において実行される直接実行形式のプログラムを生成することを特徴とする請求項１に記載のコンパイル処理装置。

【請求項６】

前記第１の実行形式プログラム生成部は、間接実行形式のプログラムを生成することを特徴とする請求項１に記載のコンパイル処理装置。

【請求項７】

前記第１の実行形式プログラム生成部は、プログラムを実行する計算機に依存することなく実行可能な間接実行形式の中間プログラムを生成する間接実行形式中間プログラム生成部であることを特徴とする請求項６に記載のコンパイル処理装置。

【請求項８】

操作者からのコンパイル指示情報の入力を受け付けるコンパイル指示情報受付部を備え、

前記第２の実行形式プログラム生成部は、前記コンパイル指示情報受付部を通じて入力された前記コンパイル指示情報と、前記解析情報と、前記プロファイル情報とに基づいて、目的とする計算機の最適化された実行形式のプログラムを生成することを特徴とする請求項１に記載のコンパイル処理装置。

【請求項９】

操作者からのコンパイル指示情報の入力を受け付けるコンパイル指示情報受付部と、

前記コンパイル指示情報受付部を通じて入力された前記コンパイル指示情報を記憶する操作者用のコンパイル指示情報記憶部と

を備え、

前記第２の実行形式プログラム生成部は、前記コンパイル指示情報記憶部の前記コンパイル指示情報と、前記解析情報と、前記プロファイル情報とに基づいて、目的とする計算機の最適化された実行形式のプログラムを生成することを特徴とする請求項１に記載のコンパイル処理装置。

【請求項１０】

コンパイル処理を補助するためのプログラムからコンパイル指示情報を受け付けるコンパイル指示情報受付部を備え、

前記第２の実行形式プログラム生成部は、前記コンパイル指示情報受付部を通

じて入力された前記コンパイル指示情報と、前記解析情報と、前記プロファイル情報とに基づいて、目的とする計算機の最適化された実行形式のプログラムを生成することを特徴とする請求項１に記載のコンパイル処理装置。

【請求項１１】

コンパイル処理を補助するためのプログラムからのコンパイル指示情報を受け付けるコンパイル指示情報受付部と、

前記コンパイル指示情報受付部を通じて受け付けた前記コンパイル指示情報を記憶するコンパイル指示情報記憶部と

を備え、

前記第２の実行形式プログラム生成部は、前記コンパイル指示情報記憶部の前記コンパイル指示情報と、前記解析情報と、前記プロファイル情報とに基づいて、目的とする計算機の最適化された実行形式のプログラムを生成することを特徴とする請求項１に記載のコンパイル処理装置。

【請求項１２】

ソースプログラムから、そのソースプログラムの解析情報を生成する解析情報生成工程と、

前記解析情報に基づいて、第１の実行形式プログラムを生成する第１の実行形式プログラム生成工程と、

前記第１の実行形式プログラムに基づいて、プロファイル情報を生成するプロファイル情報生成工程と、

前記解析情報と、前記プロファイル情報とに基づいて、第２の実行形式プログラムを生成する第２の実行形式プログラム生成工程と

を有するコンパイル処理方法。

【請求項１３】

前記解析情報生成工程により生成される前記解析情報を、解析情報記憶部に記録する解析情報記録工程を有することを特徴とする請求項１２に記載のコンパイル処理方法。

【請求項１４】

前記プロファイル情報生成工程により生成される前記プロファイル情報をプロ

ファイル情報記憶部に記録するプロファイル情報記録工程を有することを特徴とする請求項１２に記載のコンパイル処理方法。

【請求項１５】

前記第１の実行形式プログラム生成工程は、目的とする計算機において実行される直接実行形式のプログラムを生成することを特徴とする請求項１２に記載のコンパイル処理方法。

【請求項１６】

前記第１の実行形式プログラム生成工程は、間接実行形式のプログラムを生成することを特徴とする請求項１２に記載のコンパイル処理方法。

【請求項１７】

前記第１の実行形式プログラム生成工程は、プログラムを実行する計算機に依存することなく実行可能な間接実行形式の中間プログラムを生成する間接実行形式中間プログラム生成工程であることを特徴とする請求項１６に記載のコンパイル処理方法。

【請求項１８】

操作者からのコンパイル指示情報の入力を受け付けるコンパイル指示情報受付工程を備え、

前記第２の実行形式プログラム生成工程においては、前記コンパイル指示情報受付工程において受け付けられた前記コンパイル指示情報と、前記解析情報と、前記プロファイル情報とに基づいて、目的とする計算機の最適化された実行形式のプログラムを生成することを特徴とする請求項１２に記載のコンパイル処理方法。

【請求項１９】

操作者からのコンパイル指示情報の入力を受け付けて、これをコンパイル指示情報記憶部に記憶するコンパイル指示情報受付工程を備え、

前記第２の実行形式プログラム生成工程においては、前記コンパイル指示情報記憶部の前記コンパイル指示情報と、前記解析情報と、前記プロファイル情報とに基づいて、目的とする計算機の最適化された実行形式のプログラムを生成することを特徴とする請求項１２に記載のコンパイル処理方法。

【請求項２０】

コンパイル処理を補助するためのプログラムからコンパイル指示情報を受け付けるコンパイル指示情報受付工程を備え、

前記第２の実行形式プログラム生成工程においては、前記コンパイル指示情報受付工程において受け付けられた前記コンパイル指示情報と、前記解析情報と、前記プロファイル情報とに基づいて、目的とする計算機の最適化された実行形式のプログラムを生成することを特徴とする請求項１２に記載のコンパイル処理方法。

【請求項２１】

コンパイル処理を補助するためのプログラムからのコンパイル指示情報を受け付けて、これをコンパイル指示情報記憶部に記憶するコンパイル指示情報受付工程を備え、

前記第２の実行形式プログラム生成工程においては、前記コンパイル指示情報記憶部の前記コンパイル指示情報と、前記解析情報と、前記プロファイル情報とに基づいて、目的とする計算機の実行形式のプログラムを生成することを特徴とする請求項１２に記載のコンパイル処理方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

この発明は、例えば、逐次方式、並列方式、あるいは、ＶＬＩＷ（Ｖｅｒｙ Ｌｏｎｇ Ｉｎｓｔｒｕｃｔｉｏｎ Ｗｏｒｄ）方式などの計算機上でプログラミング言語で記述されたプログラムから実行形式のプログラムを生成するコンパイル処理装置に関する。

【０００２】

【従来の技術】

プログラミング言語で記述されたソースプログラムから、目的の電子計算機において実行可能な形式のプログラム（以下、この明細書においては実行形式プログラムという）を生成したり、最適化された実行形式プログラムを生成する場合に、コンパイル処理装置が用いられる。

【０００３】

例えば、図１６に示すように、プログラマによって作成され、ソースプログラム記憶部１０１に記憶されたプログラミング言語で記載されたソースプログラムは、コンパイル処理装置１０２に入力される。コンパイル処理装置１０２は、ソースプログラムの字句解析、構文解析、意味解析などを行って、ソースプログラムから目的の電子計算機で実行可能な実行形式プログラムを生成し、これを実行形式プログラム記憶部１０３に記憶する。

【０００４】

この実行形式プログラム記憶部１０３に記憶された実行形式プログラムを実行することにより得られる情報から、その実行形式プログラムのプロファイル情報が得られ、このプロファイル情報がプロファイル情報記憶部１０４に記憶される。プロファイル情報は、例えば、実行されたプログラムのどの部分が何回実行され、その実行時間がどれくらいであったかなど、プログラムを実行することにより得られるプログラムの動作に関する様々な情報を含んでいる。

【０００５】

そして、図１６に示すように、コンパイル処理装置１０２に、ソースプログラム記憶部１０１に記憶されているプログラミング言語で記載されたソースプログラムと、プロファイル情報記憶部１０４に記憶されたそのプログラムに対するプロファイル情報とを入力して、再度コンパイル処理を実行することにより、プログラムの最適化を行って、最適化され、高速化された実行形式プログラムが生成され、これが実行形式プログラム記憶部１０３に記憶される。

【０００６】

【発明が解決しようとする課題】

ところで、図１６を用いて前述したように、最適化された実行形式プログラムを生成する場合、まず、ソースプログラムをコンパイル処理装置に入力して、目的とする電子計算機の実行形式プログラムを生成し、この実行形式プログラムを実行することにより得られる情報から、プロファイル情報を得る。

【０００７】

そして、コンパイル処理装置に入力したソースプログラムと、プロファイル情

報とをコンパイル処理装置に入力し、再度コンパイル処理を行うことにより、最適化された実行形式プログラムが生成される。

【０００８】

このため、プロファイル情報を得るために実行形式プログラムを生成するための最初のコンパイル処理と最適化されたプログラムを得るための再度のコンパイル処理において、ソースプログラムの字句解析、構文解析、意味解析などの解析処理が重複して行われるなどの無駄が生じ、最適化された実行形式プログラムを生成するまでに時間がかかってしまうという問題がある。

【０００９】

以上のことにかんがみ、この発明は、処理時間を短縮し、迅速に実行形式プログラムを生成することができるコンパイル処理装置およびコンパイル処理方法を提供することを目的とする。

【００１０】

【課題を解決するための手段】

上記課題を解決するため、請求項１に記載の発明のコンパイル処理装置は、

ソースプログラムから、そのソースプログラムの解析情報を生成する解析情報生成部と、

前記解析情報に基づいて、第１の実行形式プログラムを生成する第１の実行形式プログラム生成部と、

前記第１の実行形式プログラムに基づいて、プロファイル情報を生成するプロファイル情報生成部と、

前記前記解析情報と、前記プロファイル情報とに基づいて、第２の実行形式プログラムを生成する第２の実行形式プログラム生成部と

を備えることを特徴とする。

【００１１】

この請求項１に記載の発明のコンパイル処理装置によれば、解析情報生成部により、ソースプログラムに対する字句解析、構文解析、意味解析などの種々の解析処理が行われて、ソースプログラムの解析情報が生成される。この解析情報に基づいて、第１の実行形式プログラム生成部により、第１の実行形式プログラム

が生成される。

【００１２】

そして、第１の実行形式プログラムを実行することにより得られる情報から、プロファイル情報生成部によりプロファイル情報が生成される。このプロファイル情報と、解析情報生成部により生成された解析情報とに基づいて、第２の実行形式プログラム生成部により第２の実行形式のプログラムが生成される。

【００１３】

これにより、ソースプログラムを繰り返し用いることなく、ソースプログラムを解析処理することにより生成されたソースプログラムの解析情報を用いて、第１の実行形式プログラム、第２の実行形式プログラムが生成される。したがって、ソースプログラムについての解析処理を重複して行う必要がなくなり、迅速に最適化した実行形式プログラムを生成することができるようにされる。

【００１４】

また、請求項２に記載の発明のコンパイル処理装置は、請求項１に記載のコンパイル処理装置であって、

前記解析情報生成部により生成される前記解析情報を記憶する解析情報記憶部を備えることを特徴とする。

【００１５】

この請求項２に記載のコンパイル処理装置によれば、解析情報生成部により生成される解析情報が、解析情報記憶部に記憶され保持するようにされる。これにより、第１の実行形式プログラムを生成する場合にも、また、第２の実行形式プログラムを生成する場合にも、解析情報記憶部の解析情報を用いることができるので、ソースプログラムの解析処理を繰り返し行うことなく、迅速に、第１、第２の実行形式プログラムを繰り返し生成することができるようにされる。

【００１６】

また、請求項３に記載の発明のコンパイル処理装置は、請求項１に記載のコンパイル処理装置であって、

前記プロファイル情報生成部により生成される前記プロファイル情報を記憶するプロファイル情報記憶部を備えることを特徴とする。

【００１７】

この請求項３に記載のコンパイル処理装置によれば、プロファイル情報生成部により生成されるプロファイル情報が、プロファイル情報記憶部に記憶され保持するようにされる。これにより、１度生成したプロファイル情報を用いて、第２の実行形式プログラムを繰り返し生成することができるようにされる。

【００１８】

また、請求項４に記載の発明のコンパイル処理装置は、請求項１に記載のコンパイル処理装置であって、

前記第１の実行形式プログラム生成部と、前記第２の実行形式プログラム生成部とは、１つの実行形式プログラム生成部により構成されることを特徴とする。

【００１９】

この請求項４に記載の発明のコンパイル処理装置によれば、第１の実行形式プログラムも、第２の実行形式のプログラムも同じ１つの実行形式プログラム生成部により生成するようにされる。つまり、第１の実行形式プログラム生成部と、第２の実行形式プログラム生成部とは、共通に用いられる。

【００２０】

これにより、プロファイル情報を得るために実行するようにされる第１の実行形式プログラムも、プロファイル情報をも考慮した第２の実行形式プログラムも、同じ１つの実行形式プログラム生成部により生成することができるようにされる。したがって、第１の実行形式プログラム生成部と、第２の実行形式プログラム生成部とを別々に設ける必要がなくなり、コンパイル処理装置の構成を簡単にすることができる。

【００２１】

また、請求項５に記載の発明のコンパイル処理装置は、請求項１に記載のコンパイル処理装置であって、

前記第１の実行形式プログラム生成部は、目的とする計算機において実行される直接実行形式のプログラムを生成することを特徴とする。

【００２２】

この請求項５に記載の発明のコンパイル処理装置によれば、第１の実行形式プ

ログラム生成部より、目的とする計算機で実行される直接実行形式のプログラムが生成される。ここで、直接実行形式のプログラムは、そのプログラムを実行する目的とする計算機に固有の形式の例えば機械語のプログラムである。

【００２３】

これにより、このコンパイル処理装置において、目的とする計算機において実行可能であるその計算機に固有な形式であって、その目的とする計算機のプログラムの実行環境などに適合したプログラムを生成することができるようにされる。そして、この直接実行形式のプログラムをコンパイル処理装置を有する装置において実行することにより、その直接実行形式のプログラムのプロファイル情報を得て、最適化された実行形式のプログラムを生成することができるようにされる。

【００２４】

また、請求項６に記載の発明のコンパイル処理装置は、請求項１に記載のコンパイル処理装置であって、

前記第１の実行形式プログラム生成部は、間接実行形式のプログラムを生成することを特徴とする。

【００２５】

この請求項６に記載の発明のコンパイル処理装置によれば、第１の実行形式プログラム生成部より、間接実行形式のプログラムが生成される。ここで、間接実行形式のプログラムは、前述した直接実行形式以外の形式のプログラムであり、例えば、米国において開発されたプログラミング言語であるＪａｖａ（ジャバ）のバイトコードのような、計算機に依存することなく実行可能なものである。

【００２６】

これにより、このコンパイル装置において、間接実行形式のプログラムが生成できるようにされる。間接実行形式のプログラムは、これを実行する計算機などの機器に依存することなく実行することができるので、コンパイル処理装置において、間接実行形式のプログラムを実行することにより、プロファイル情報を得て、このプロファイル情報をも用いることによって、第２の実行形式のプログラムを作成することができるようにされる。

【００２７】

また、請求項７に記載の発明のコンパイル処理装置は、請求項６に記載のコンパイル処理装置であって、

前記第１の実行形式プログラム生成部は、プログラムを実行する計算機に依存することなく実行可能な間接実行形式の中間プログラムを生成する間接実行形式中間プログラム生成部であることを特徴とする。

【００２８】

この請求項７に記載の発明のコンパイル処理装置によれば、間接実行形式中間プログラム生成部により、プログラムを実行する計算機のプログラムの実行環境などに依存することなく実行可能な間接実行形式のいわゆる中間コードプログラムが生成され、この生成されたプログラムが、コンパイル処理を行う自装置において実行可能にされる。そして、間接実行形式中間プログラム生成部により生成された間接実行形式のいわゆる中間コードプログラムを実行することにより得られる情報から、プロファイル情報が生成される。

【００２９】

これにより、プログラムが実行される計算機のプログラムの実行環境などに依存することなく、間接実行形式中間プログラム生成部により生成されるいわゆる中間コードプログラムを実行することにより、そのプログラムに対するプロファイル情報を得て、このプロファイル情報を用いることにより、目的とする間接実行形式の最適化されたプログラムを簡単かつ迅速に生成することができるようにされる。

【００３０】

また、請求項８に記載の発明のコンパイル処理装置は、請求項１に記載のコンパイル処理装置であって、

操作者からのコンパイル指示情報の入力を受け付けるコンパイル指示情報受付部を備え、

前記第２の実行形式プログラム生成部は、前記コンパイル指示情報受付部を通じて入力された前記コンパイル指示情報と、前記解析情報と、前記プロファイル情報とに基づいて、目的とする計算機の最適化された実行形式のプログラムを生

成することを特徴とする。

【００３１】

この請求項８に記載の発明のコンパイル処理装置によれば、コンパイル指示情報受付部を通じて、操作者から、例えば、直接実行形式、間接実行形式の別や、コンパイル処理により生成されるプログラムを実行する目的とする計算機のプログラムの実行環境などの種々のコンパイル指示情報が、コンパイル指示情報受付部を通じて受け付けられる。

【００３２】

このコンパイル指示情報受付部を通じて受け付けられたコンパイル指示情報が、第２の実行形式プログラム生成部に供給される。そして、第２の実行形式プログラム生成部により、解析情報、プロファイル情報、および、操作者からのコンパイル指示情報に基づいて、目的とする計算機で実行可能とされた実行形式プログラムが生成される。

【００３３】

これにより、操作者からのコンパイル指示情報をも考慮して、第２の実行形式プログラムが生成されるので、操作者からの要求に応じた実行形式プログラムを簡単かつ迅速に生成することができるようにされる。

【００３４】

また、請求項９に記載の発明のコンパイル処理装置は、請求項１に記載のコンパイル処理装置であって、

操作者からのコンパイル指示情報の入力を受け付けるコンパイル指示情報受付部と、

前記コンパイル指示情報受付部を通じて入力された前記コンパイル指示情報を記憶する操作者用のコンパイル指示情報記憶部と

を備え、

前記第２の実行形式プログラム生成部は、前記コンパイル指示情報記憶部の前記コンパイル指示情報と、前記解析情報と、前記プロファイル情報とに基づいて、目的とする計算機の最適化された実行形式のプログラムを生成することを特徴とする。

【００３５】

この請求項９に記載の発明のコンパイル処理装置によれば、コンパイル指示情報受付部を通じて、操作者から入力された種々のコンパイル指示情報が、コンパイル指示情報記憶部に記憶される。

【００３６】

そして、第２の実行形式プログラム生成部により、解析情報、プロファイル情報、および、コンパイル指示情報記憶部に記憶された操作者からのコンパイル指示情報に基づいて、第２の実行形式プログラムが生成される。

【００３７】

これにより、操作者は、コンパイル処理前に予め決まっているコンパイル指示情報について、予めコンパイル指示情報記憶部に記憶させておくことができるようにされる。したがって、コンパイル処理実行時のコンパイル指示情報の入力処理にかかる時間を短縮することができ、コンパイル指示情報記憶部に記憶されたコンパイル指示情報をも考慮した、操作者からの要求に応じた実行形式プログラムをより迅速に生成することができるようにされる。

【００３８】

また、請求項１０に記載の発明のコンパイル処理装置は、請求項１に記載のコンパイル処理装置であって、

コンパイル処理を補助するためのプログラムからコンパイル指示情報を受け付けるコンパイル指示情報受付部を備え、

前記第２の実行形式プログラム生成部は、前記コンパイル指示情報受付部を通じて入力された前記コンパイル指示情報と、前記解析情報と、前記プロファイル情報とに基づいて、目的とする計算機の最適化された実行形式のプログラムを生成することを特徴とする。

【００３９】

この請求項８に記載の発明のコンパイル処理装置によれば、コンパイル指示情報受付部を通じて、コンパイル処理を補助するためのプログラムから提供される。例えば、コンパイル処理に必要なパラメータなどの種々のコンパイル指示情報が、コンパイル指示情報受付部を通じて受け付けられ、これが、第２の実行形式

プログラム生成部に供給される。

【００４０】

そして、第２の実行形式プログラム生成部により、解析情報、プロファイル情報、および、コンパイル処理を補助するためのプログラムからのコンパイル指示情報に基づいて、目的とする計算機で実行可能とされた第２の実行形式プログラムが生成される。

【００４１】

これにより、コンパイル処理を補助するためのプログラムからのコンパイル指示情報をも考慮して、第２の実行形式プログラムが生成されるので、目的とする第２の実行形式プログラムを簡単かつ迅速に生成することができるようにされる。

【００４２】

また、請求項１１に記載の発明のコンパイル処理装置は、請求項１に記載のコンパイル処理装置であって、

コンパイル処理を補助するためのプログラムからのコンパイル指示情報を受け付けるコンパイル指示情報受付部と、

前記コンパイル指示情報受付部を通じて受け付けた前記コンパイル指示情報を記憶するコンパイル指示情報記憶部と

を備え、

前記第２の実行形式プログラム生成部は、前記コンパイル指示情報記憶部の前記コンパイル指示情報と、前記解析情報と、前記プロファイル情報とに基づいて、目的とする計算機の最適化された実行形式のプログラムを生成することを特徴とする。

【００４３】

この請求項１１に記載の発明のコンパイル処理装置によれば、コンパイル指示情報受付部を通じて、コンパイル処理を補助するためのプログラムから提供された種々のコンパイル指示情報が、コンパイル指示情報記憶部に記憶される。

【００４４】

そして、第２の実行形式プログラム生成部により、解析情報、プロファイル情

報、および、コンパイル指示情報記憶部に記憶されたコンパイル処理を補助するためのプログラムからのコンパイル指示情報に基づいて、目的とする計算機で実行可能とされた第２の実行形式プログラムが生成される。

【００４５】

これにより、コンパイル処理に必要なコンパイル処理を補助するプログラムからの種々のコンパイル指示情報を予めコンパイル指示情報記憶部に記憶させておくことができるので、コンパイル処理時にコンパイル処理を補助するためのプログラムを実行させる必要がなくなり、コンパイル指示情報記憶部に記憶されたコンパイル指示情報をも考慮した、目的とする実行形式プログラムをより迅速に生成することができるようにされる。

【００４６】

【発明の実施の形態】

以下、図を参照しながらこの発明によるコンパイル処理装置およびコンパイル処理方法の一実施の形態について説明する。

【００４７】

［第１の実施の形態］

図１は、この第１の実施の形態のコンパイル装置１を説明するための図である。図１に示すように、この第１の実施の形態のコンパイル処理装置１は、ソースプログラム記憶部１０に記憶されている高水準プログラミング言語で記述されたソースプログラムから、目的の電子計算機において実行可能な実行形式プログラムを生成し、これを実行形式プログラム記憶部２０に記憶するものである。

【００４８】

さらに、コンパイル処理装置１は、実行形式プログラム記憶部２０に記憶された実行形式プログラムを実行することにより得られる情報からプロファイル情報を生成し、このプロファイル情報を用いて、実行形式プログラムを生成する処理を行うことにより、最適化された実行形式プログラムを生成することができるものである。

【００４９】

この第１の実施の形態のコンパイル処理装置１について説明する。この第１の

実施の形態のコンパイル処理装置１は、図１に示すように、ソースプログラム解析部１１、ソースプログラム解析情報記憶部１２、実行形式生成部１３、プロファイル情報生成部１４、プロファイル情報記憶部１５を備えている。

【００５０】

ソースプログラム記憶部１０に記憶されている高水準プログラミング言語で記述されたソースプログラムは、コンパイル処理装置１のソースプログラム解析部１１に供給される。ソースプログラム解析部１１は、解析情報生成部としての機能を有しており、ソースプログラムの文字解析、構文解析、意味解析などの解析処理を行って、ソースプログラムからそのソースプログラムの解析情報を生成し、これをソースプログラム解析情報記憶部１２に記憶する。

【００５１】

このソースプログラム解析情報記憶部１２に記憶されるソースプログラムの解析情報は、ソースプログラムを解析することにより生成された、例えば、中間コード、アセンブリコード、機械語などの生成情報を含むものであり、実行形式プログラムを生成するために必要な情報を含むものである。

【００５２】

そして、実行形式生成部１３は、ソースプログラム解析情報記憶部１２に記憶された解析情報から、目的とする電子計算機において実行される実行形式プログラムを生成し、これを実行形式プログラム記憶部２０に記憶する。これによって、目的とする電子計算機において実行可能な実行形式プログラムが、実行形式プログラム記憶部２０に得られる。

【００５３】

しかし、この段階の実行形式プログラムは、ソースプログラムからそのまま生成されたものであり、最適化されていない。このため、この実行形式プログラムを実行しても、無駄な処理が行われ、高速な処理が行えない場合もある。

【００５４】

そこで、コンパイル処理装置１のプロファイル情報生成部１４は、実行形式プログラム記憶部２０に記憶されている目的とする実行形式プログラムを実行することにより得られる情報から、プロファイル情報を生成し、これをプロファイル

情報記憶部１５に記憶する。

【００５５】

そして、コンパイル処理装置１において、ソースプログラム解析情報記憶部１２に記憶されている最初のコンパイル処理によって生成されたソースプログラムの解析情報と、プロファイル情報記憶部１５に記憶されているプロファイル情報とを用いて、実行形式生成部１３により、再度の実行形式プログラムの生成処理を行う。

【００５６】

この再度の実行形式プログラムの生成処理は、最適化された実行形式プログラムを生成するための処理である。この処理は、図１に示すように、ソースプログラム解析情報記憶部１２に記憶された解析情報から生成されるプログラムに対して、プロファイル情報記憶部１５のプロファイル情報を用いて最適化を行い、無駄な動作を行うことなく高速化された実行形式プログラムを生成し、これを実行形式プログラム記憶部２０に記憶するものである。

【００５７】

図２は、この実施の形態のコンパイル処理装置１の実行形式生成部１３を説明するための図である。この実施の形態の実行形式生成部１３は、図２に示すように、プログラム構築部１３１と、最適化した実行形式プログラムを生成するために、プロファイル情報からプログラム実行についての確率情報を生成する確率情報生成部１３２を備えている。

【００５８】

前述したように、最初（１回目）のコンパイル処理においては、目的とするプログラムについてのプロファイル情報がまだプロファイル情報記憶部１５に生成されていない。このため、実行形式生成部１３は、目的とするソースプログラムについての最初のコンパイル処理においては、ソースプログラム解析情報記憶部１２に記憶された中間コードやアセンブリコードなどの解析情報のみから、例えば、目的とする電子計算機において実行可能な実行形式プログラムを生成し、これを実行形式プログラム記憶部２０に記憶する。

【００５９】

そして、前述したように、コンパイル処理装置１のプロファイル情報生成部１４は、実行形式プログラム記憶部２０に記憶されている最適化前の実行形式プログラムを実行することにより得られる情報から、プロファイル情報を生成し、これをプロファイル情報記憶部１５に記憶する。

【００６０】

プロファイル情報記憶部１５に記憶されるプロファイル情報は、実行形式プログラムの実行を開始してからの経過時間と、各経過時間における詳細情報とからなるものである。そして、前述もしたように、プロファイル情報記憶部１５に記憶されたプロファイル情報を用いて再度の実行形式プログラムの生成処理が行われることにより、最適化された実行形式プログラムが生成される。

【００６１】

この場合には、ソースプログラム解析情報記憶部１２の中間コードやアセンブリコードなどの解析情報が、プログラム構築部１３１に供給されるとともに、プログラム実行の確率情報生成部１３２により、プロファイル情報記憶部１５に記憶されているプロファイル情報を統計処理することにより生成されるプログラム実行の確率情報が、プログラム構築部１３１に供給される。

【００６２】

実行形式生成部１３の確率情報生成部１３２により生成される確率情報は、後述もするが、例えば、プログラムを構成するブロックの実行回数、各命令の実行回数、分岐の回数、メモリやレジスタのアクセス回数などの各種の確率情報である。

【００６３】

そして、実行形式生成部１３のプログラム構築部１３１は、ソースプログラム解析情報記憶部１２からの解析情報から生成されるプログラムに対して、プロファイル情報記憶部１５のプロファイル情報から生成された確率情報を用いて最適化を行い、無駄な動作を行うことなく高速化された実行形式プログラムを生成し、これを実行形式プログラム記憶部２０に記憶する。

【００６４】

この場合、上述のように、ソースプログラム解析部１１により各種の解析処理が行われて生成され、ソースプログラム解析情報記憶部１２に記憶された解析情報が用いられて、最適化された実行形式プログラムが生成される。すなわち、最適化された実行形式プログラムを生成するために、従来のように、ソースプログラムの解析処理を繰り返すことがないので、最適化された実行形式プログラムを生成するための処理にかかる時間を短縮し、迅速に最適化された実行形式プログラムを生成することができる。

【００６５】

そして、この第１の実施の形態のコンパイル処理装置１において行なわれる最適化された実行形式プログラムの生成手順は、以下に説明するように、第１、第２の２つのプロセスからなっている。

【００６６】

すなわち、この第１の実施の形態のコンパイル処理装置１においては、まず、ソースプログラム記憶部１０からソースプログラムを読み出し、そのソースプログラムをソースプログラム解析部１１において字句解析、構文解析、意味解析などの解析処理を行うことにより、ソースプログラムの解析情報を生成し、これをソースプログラム解析情報記憶部１２に記憶する（図１①）。

【００６７】

そして、ソースプログラム解析情報記憶部１２に記憶された解析情報から、プロファイル情報を得るために実行する実行形式プログラムを生成する（図１②）。このプロファイル情報を得るためにソースプログラムから実行形式プログラムを生成するまでの処理が第１のプロセスである。

【００６８】

この第１のプロセスで生成された実行形式プログラムを実行することにより得られる情報に基づいて、プロファイル情報生成部１４によりプロファイル情報を生成し（図１③）、これをプロファイル情報記憶部１５に記憶する（図１④）。

【００６９】

そして、第１のプロセスにおいて生成され、ソースプログラム解析情報記憶部

１２に記憶されている解析情報と、プロファイル情報記憶部１５に記憶されたプロファイル情報とに基づいて、実行形式生成部１３において、再度の実行形式プログラムの生成を行うことにより、プロファイル情報に基づいて最適化された実行形式プログラムが生成され、これが実行形式プログラム記憶部２０に記憶される（図１⑤）。このプロファイル情報の生成から、実行形式生成部１３による再度の実行形式プログラムの生成までが第２のプロセスとなる。

【００７０】

そして、第２のプロセスにおいては、ソースプログラム解析情報記憶部１２に記憶されている解析情報と、プロファイル情報とを用いて最適化した実行形式プログラムを生成するので、ソースプログラムの解析処理を行わなくても済む。

【００７１】

すなわち、第１のプロセスにおいて、ソースプログラム解析部１１によりソースプログラムの解析処理を行った後においては、第２のプロセスにおいて、ソースプログラムの解析処理を繰り返し行う必要がないので、最適化された実行形式プログラムを生成するまでにかかる時間を短縮し、最適化された実行形式プログラムを迅速に生成することができる。

【００７２】

次に、この第１の実施の形態のコンパイル処理装置１においておこなわれるコンパイル処理について、具体例を用いてより詳細に説明する。

図３は、ソースプログラム記憶部１０に記憶されるソースプログラムの例を説明するための図である。この例のソースプログラムは、条件により変数ａまたは変数ｂのいづれかを選択し、選択した変数の値を２乗したものを変数ｄに格納するというものである。

【００７３】

そして、前述したように、ソースプログラム記憶部１０に記憶されているソースプログラムを、ソースプログラム解析部１１により解析することにより生成される、中間コード、アセンブリコード、機械語などの解析情報（生成情報）が、ソースプログラム解析情報記憶部１２に記憶される。この場合、中間コード、アセンブリコード、機械語などのうちいずれを生成するかは、例えば、ユーザによ

り選択することができるようにされる。

【００７４】

図４、図５は、ソースプログラム記憶部１０に記憶されている図３に示したソースプログラムがソースプログラム解析部１１により解析されて、ソースプログラム解析情報記憶部１２に生成される解析情報の例を説明するための図である。このうち、図４は、生成された解析情報が中間コードの場合の例を概念的に示し、図５は、生成された解析情報がアセンブリコードの場合の例を示している。

【００７５】

この例において、中間コード（図４）のＢＬＫ１とアセンブリコード（図５）のｂｌｏｃｋ１とは、変数ａと変数ｂとを比較するブロックである。また、中間コード（図４）のＢＬＫ２とアセンブリコード（図５）のｂｌｏｃｋ２とは、変数ａが変数ｂより大きい場合に、変数ｃに変数ａを格納するブロックであり、中間コード（図４）のＢＬＫ３とアセンブリコード（図５）のｂｌｏｃｋ３とは、変数ｂが変数ａ以上である場合に、変数ｃに変数ｂの値を格納するブロックである。そして、中間コード（図４）のＢＬＫ４とアセンブリコード（図５）のｂｌｏｃｋ４とは、選択された変数を２乗してその結果を変数ｄに格納するブロックである。

【００７６】

なお、図５において、「ｌｄ」は、ロード命令、「ｓｔ」は、ストア命令、「ｃｍｐ」は、比較命令、「ｂｌｅ」は、分岐命令、「ｊｍｐ」は、飛び越し命令、「ｍｕｌ」は、掛け算命令である。したがって、図５に示すこの実施の形態のアセンブリコードの例の場合には、まず、ｂｌｏｃｋ１の第１ステップにおいて、変数ａがレジスタｒ１に格納され、第２ステップにおいて、変数ｂがレジスタｒ２に格納される。

【００７７】

そして、ｂｌｏｃｋ１の第３ステップにおいて、レジスタｒ１とレジスタｒ２との値が比較され、比較結果がレジスタｒ３に格納される。そして、ｂｌｏｃｋ１の第４ステップにおいて、レジスタｒ３に格納された比較結果が、ｒ１≦ｒ２（変数ａ≦変数ｂ）であれば、ｂｌｏｃｋ３への飛び越しが行われる。したがっ

て、レジスタｒ３に格納された比較結果が、ｒ１＞ｒ２（変数ａ＞変数ｂ）であれば、ｂｌｏｃｋ２の処理が行われることになる。

【００７８】

ｂｌｏｃｋ２は、上述のようにｂｌｏｃｋ１においてのレジスタｒ１の値とレジスタｒ２の値との比較結果が、ｒ１＞ｒ２である場合に行われる処理である。このｂｌｏｃｋ２においては、第１ステップにおいて、変数ａがレジスタｒ４に格納され、第２ステップにおいて、レジスタｒ４に格納されている値が、変数ｃに格納される。そして、ｂｌｏｃｋ２の第３ステップにより、ｂｌｏｃｋ４への飛び越しが行われる。

【００７９】

ｂｌｏｃｋ３は、上述のようにｂｌｏｃｋ１においてのレジスタｒ１の値とレジスタｒ２の値との比較結果が、ｒ１≦ｒ２である場合に行われる処理である。このｂｌｏｃｋ３においては、まず、第１ステップにおいて、変数ｂが、レジスタｒ５に格納され、第２ステップにおいて、レジスタｒ５の値が、変数ｃに格納される。そして、ｂｌｏｃｋ３の第３ステップにより、ｂｌｏｃｋ４への飛び越しが行われる。

【００８０】

そして、ｂｌｏｃｋ４においては、第１ステップにおいて、変数ｃがレジスタｒ６に格納され、第２ステップにおいて、さらに変数ｃが、レジスタｒ７に格納される。そして、ｂｌｏｃｋ４の第３ステップにおいて、レジスタｒ６に格納されている値と、レジスタｒ７に格納されている値との掛け算が行われ、第４ステップにおいて、第３ステップの掛け算の結果が、レジスタｒ８に格納される。

【００８１】

このように、図５に示したこの実施の形態で用いるソースプログラムのアセンブリコードは、変数ａと変数ｂの大小関係に応じて、変数ａの２乗、あるいは、変数ｂの２乗を求め、レジスタｒ８に格納するようにしたものである。

【００８２】

そして、実行形式生成部１３は、前述のようにソースプログラム解析情報記憶部１２に記憶された中間コードやアセンブリコードなどの解析情報から、目的と

する電子機器において実行される形式の実行形式プログラムを生成し、これを実行形式プログラム記憶部２０に記憶する。ここで、実行形式プログラム記憶部２０に生成された実行形式プログラムは、最初（１回目）のコンパイル処理により生成された実行形式プログラムであり、最適化前の実行形式プログラムである。

【００８３】

次に、コンパイル処理装置１のプロファイル情報生成部１４は、実行形式プログラム記憶部２０に記憶されている目的とする実行形式プログラムを実行することにより得られる情報から、プロファイル情報を生成し、これをプロファイル情報記憶部１５に記憶する。

【００８４】

図６は、プロファイル情報生成部１４により、プロファイル情報記憶部１５に生成されるプロファイル情報の例を説明するための図である。この図６に示すプロファイル情報は、例えば、図５に示したアセンブリコードから生成された実行形式プログラムを実行することにより得られる情報から生成されるものである。

【００８５】

図６に示すように、プロファイル情報記憶部１５に記憶されるプロファイル情報は、実行形式プログラムの実行を開始してからの経過時間と、各経過時間における詳細情報とからなるものである。すなわち、プロファイル情報は、実行形式プログラム記憶部２０に記憶されている実行形式プログラムを実行した場合に、時間の経過とともに、どの命令が実行され、どこで分岐し、また、どのメモリがアクセスされたか、あるいは、どのレジスタがアクセスされたかなど、プログラムの実行の状態を示すものである。

【００８６】

この図６に示したプロファイル情報の例は、以下のように、実行形式プログラムの実行を開始してからの各経過時間における詳細情報が示されている。つまり、この図６の例の場合には、経過時間１００５０において、プログラムカウンタ（ＰＣ）が変化し、図５に示したアセンブリコードのｂｌｏｃｋ１が実行されて、変数ａがレジスタに格納（ロード）されている。そして、経過時間１００５１において、変数ｂがレジスタに格納（ロード）され、経過時間１００５２におい

て、比較命令（ｃｏｍｐａｒｅ）が実行されている。

【００８７】

経過時間１００５３において、分岐命令（ｂｒａｎｃｈ　ｌｅｓｓ　ｏｒ　ｅｑｕａｌ）が実行されて、図５に示したアセンブリコードのｂｌｏｃｋ３を実行するようにされている。そして、経過時間１００５４において、プログラムカウンタ（ＰＣ）が変化し、図５に示したアセンブリコードのｂｌｏｃｋ３が実行されて、変数ｂがレジスタに格納（ロード）されている。そして、経過時間１００５５において、レジスタの値が変数ｃに格納（ストア）され、経過時間１００５６において、図５に示したアセンブリコードのｂｌｏｃｋ４を実行するように飛び越し命令（ｊｕｍｐ）が実行されている。

【００８８】

この後、経過時間１００５７において、プログラムカウンタ（ＰＣ）が変化し、図５に示したアセンブリコードのｂｌｏｃｋ４が実行されて、変数ｃがレジスタに格納（ロード）され、経過時間１００５８において、さらに変数ｃの値が他のレジスタに格納（ロード）されている。

【００８９】

そして、経過時間１００５９において、掛け算命令（ｍｕｌ）が実行され、経過時間１００６０において、経過時間１００５９において実行された掛け算命令より求められた変数ｄの値が、レジスタに格納（ストア）されている。このように、実行形式プログラムを実行することにより得られた情報から、経過時間と、この経過時間に対応する実行形式プログラムの実行の状態の詳細情報が、プロファイル情報としてプロファイル情報記憶部１５に記憶される。

【００９０】

そして、コンパイル処理装置１において、ソースプログラム解析情報記憶部１２に記憶されている最初のコンパイル処理のソースプログラムの解析処理により、ソースプログラム解析情報記憶部１２に生成されたソースプログラムの解析情報と、プロファイル情報記憶部１５に記憶されているプロファイル情報とを用いて、実行形式生成部１３により、再度の実行形式プログラムの生成処理を行う。

【００９１】

この場合には、図２に示したように、ソースプログラム解析情報記憶部１２からの中間コードやアセンブリコードなどの解析情報と、確率情報生成部１３２により生成される確率情報とが、実行形式生成部１３のプログラム構築部１３１に供給される。

【００９２】

図７は、プロファイル情報記憶部１５に記憶されているプロファイル情報に基づいて、確率情報生成部１３２により生成される確率情報の例を説明するための図である。この図７に示す確率情報の例は、図５に示したアセンブリコードを実行形式プログラムに変換し、これを実行して得られる情報から形成されたプロファイル情報（図６）に基づいて生成されたものである。

【００９３】

図７に示すように、確率情報生成部１３２は、前述にもしたように、プロファイル情報記憶部１５に記憶されているプロファイル情報を統計処理することにより、各ブロックの実行回数や、各命令の実行回数、分岐の回数、メモリやレジスタのアクセス回数などの確率情報を生成する。

【００９４】

図７に示す確率情報の例の場合には、実行形式プログラムが実行されるごとに、ｂｌｏｃｋ１とｂｌｏｃｋ４とは必ず実行されるが、ｂｌｏｃｋ２が実行される確率（変数ａが変数ｂより大きい確率）は、１０％であり、ｂｌｏｃｋ３が実行される確率（変数ｂが変数ａ以上である確率）は、９０％であることを示している。

【００９５】

そして、実行形式生成部１３のプログラム構築部１３１は、ソースプログラム解析情報記憶部１２の解析情報、この例の場合にはアセンブリコードと、図７に示した確率情報とに基づいて、最適化した実行形式プログラムを生成し、これを実行形式プログラム記憶部２０に記憶する。

【００９６】

図８は、プロファイル情報から生成した確率情報を用いて最適化したプログラ

ムの例を説明するための図である。この図８は、図５に示したアセンブリコードを確率情報を用いて最適化した実行形式プログラムをアセンブリコードで示したものである。

【００９７】

図７の確率情報に示したように、ｂｌｏｃｋ３が実行される確率は９０％であり、ｂｌｏｃｋ２が実行される確率は１０％であるので、ｂｌｏｃｋ１→ｂｌｏｃｋ３→ｂｌｏｃｋ４の経路が最適化されたものが図８に示す最適化されたプログラム（アセンブリコード）である。

【００９８】

この図８に示す最適化されたアセンブリコードと、図５に示した最適化される前のアセンブリコードとを比較すると分かるように、確率情報を用いて最適化を行うと、最適化後のアセンブリコードのｂｌｏｃｋ１の３行目には、新たにレジスタｒ２の値（変数ｂ）をレジスタｒ５に格納するコードが追加され、ｂｌｏｃｋ３のコードと、ｂｌｏｃｋ４のコードが統合するようにされている。

【００９９】

すなわち、最適化が行われることにより、最適化前のｂｌｏｃｋ３の一部のコードが、最適化後のｂｌｏｃｋ１に移動され、ｂｌｏｃｋ３のコードと、ｂｌｏｃｋ４のコードが統合するようにされるとともに、レジスタの割り当ても変更されている。これにより、ｂｌｏｃｋ１において、レジスタｒ２の値（変数ｂ）がレジスタｒ５に格納されているので、変数ｂが変数ａ以上であるときには、ｂｌｏｃｋ１に続くｂｌｏｃｋ３において、変数ｃ（すなわち変数ｂ）の２乗が即座に求められ変数ｄに格納することができるようにされる。すなわち、ｂｌｏｃｋ１とｂｌｏｃｋ３の２つのｂｌｏｃｋの処理で、変数ｂの２乗を求めることができるようにされる。

【０１００】

また、変数ａが変数ｂより大きい場合には、図５に示した最適化前のアセンブリコードの場合と同様に、ｂｌｏｃｋ２において変数ａがレジスタｒ４を介して変数ｃに格納され、ｂｌｏｃｋ４において、変数ａが２乗されて変数ｄに格納されるというように、３つのブロックの処理で、変数ａの２乗を求めることができ

るようにされる。

【０１０１】

このように、プロファイル情報から生成される確率情報を考慮した最適化が行われることにより、変数ｂが変数ａ以上である確率が９０％と高いことが考慮されて、変数ｂが、変数ａ以上である場合の処理を効率よく迅速に行うことができるように、最適化した実行形式のプログラムが生成される。

【０１０２】

なお、図８においては示さなかったが、ｂｌｏｃｋ３からジャンプする先のｂｌｏｃｋ５は、ｂｌｏｃｋ４に続くブロックである。また、実行形式生成部１３は、前述したように、命令の並べ換えや追加、レジスタの割り当ての変更、分岐先の変更のほか、コードのコピー、不要なコードや命令の削除などを行って、最適化した実行形式のプログラムを生成することができるものである。

【０１０３】

このように、この実施の形態のコンパイル処理装置１の実行形式生成部１３は、最初（１回目）のコンパイル処理時においては、プロファイル情報生成用実行形式プログラム生成部として機能する。そして、プロファイル情報が生成された後においては、ソースプログラム解析情報記憶部１２のソースプログラムと、プロファイル情報記憶部１５のプロファイル情報から生成される確率情報とを用いて、最適化された実行形式プログラムを生成する最適化実行形式プログラム生成部として機能する。

【０１０４】

そして、再度の実行形式プログラムの生成処理は、最適化された実行形式プログラムを生成するための処理である。この処理は、図１に示すように、ソースプログラム解析情報記憶部１２に記憶された解析情報から生成されるプログラムに対して、プロファイル情報記憶部１５のプロファイル情報を用いて最適化を行い、無駄な動作を行うことなく高速化された実行形式プログラムを生成し、これを実行形式プログラム記憶部２０に記憶するものである。

【０１０５】

この場合、上述のように、ソースプログラム解析部１１により各種の解析処理

が行われて生成され、ソースプログラム解析情報記憶部１２に記憶された解析情報が用いられて、最適化された実行形式プログラムが生成される。すなわち、最適化された実行形式プログラムを生成するために、従来のように、ソースプログラムの解析処理を繰り返すことがないので、最適化された実行形式プログラムを生成するための処理にかかる時間を短縮し、迅速に最適化された実行形式プログラムを生成することができる。

【０１０６】

なお、実行形式プログラムには、直接実行形式と、間接実行形式の２つの実行形式があるが、この第１の実施の形態のコンパイル処理装置１の実行形式生成部１３は、直接実行形式プログラム、間接実行形式プログラムのいづれをも生成することができるものである。ここで、直接実行形式プログラムは、プログラムの実行が、プログラムを実行する機器に依存するものであり、間接実行形式プログラムは、プログラムの実行が、プログラムを実行する機器に依存することなく実行可能なものである。

【０１０７】

そして、直接実行形式プログラムを生成するか、間接実行形式プログラムを生成するかは、例えば、解析情報記憶部１２に記憶された解析情報に基づいて、自動的に切り換えることができるようにされる。例えば、解析情報記憶部１２に記憶された解析情報が、目的とする計算機で実行する実行形式プログラムを生成するためのアセンブリコードやオブジェクトコードである場合には、直接実行形式プログラムを生成し、また、解析情報記憶部１２に記憶された解析情報が、中間コードである場合には、間接実行形式プログラムを生成するというように、自動的に切り換えを行うようにすることができる。

【０１０８】

また、直接実行形式プログラムを生成するか、間接実行形式プログラムを生成するかは、例えば、コンパイル処理装置１に設けられた切り換えスイッチなどを操作することにより切り換えるようにすることもできる。

【０１０９】

同様に、ソースプログラムから解析処理をおこなって実行形式プログラムを生

成する第１のプロセスの実行形式プログラムの生成と、ソースプログラム解析情報記憶部１２に記憶された解析情報を用いて実行形式プログラムを生成する第２のプロセスの実行形式の生成とは、例えば、コンパイル処理装置が、コンパイル処理の対象になっているプログラムについて、最初のコンパイル処理か、２度目以降のコンパイルかを管理することにより、自動的に切り換えることができる。

また、例えば、コンパイル処理装置１に設けられた切り換えスイッチなどを操作することにより、第１のプロセスの実行形式プログラムの生成と、第２のプロセスの実行形式の生成とを切り換えるようにすることもできる。

【０１１０】

なお、前述の実施の形態においては、実行形式生成部１３によりプロファイル情報生成用の実行形式プログラムと、最適化された実行形式プログラムとを生成するようにしたが、これに限るものではない。つまり、プロファイル情報を生成するための実行形式生成部（プロファイル情報生成用実行形式生成部）と、最適化された実行形式プログラムを生成する最適化実行形式生成部とを別々に設けるようにしてもよい。

【０１１１】

ところで、前述にもしたように、ソースプログラム解析部１１は、中間コード、アセンブリコード、機械語などを生成することができるものである。そして、解析情報記憶部１２に記憶した解析情報が、例えば、中間コードなどの場合には、実行形式プログラムを構築しやすい形式のコードに変換して、実行形式プログラムを生成し、生成した実行形式プログラムをさらにオブジェクトコードに変換する必要が生じる。このような場合には、実行形式生成部１３に、プリコードジェネレータと、ポストコードジェネレータとが必要になる。

【０１１２】

図９は、ソースプログラム解析情報記憶部１２に生成された解析情報が、例えば、中間コードなどの場合であって、この中間コードから実行形式プログラムを生成する場合の実行形式生成部１３を説明するための図である。図９に示すように、この例の場合には、ソースプログラム解析情報記憶部１２とプログラム構築部１３１との間にプリコードジェネレータ１３３を設けるとともに、プログラム

構築部１３１の後段にポストコードジェネレータ１３４を設けたものである。

【０１１３】

プリコードジェネレータ１３３は、解析情報記憶部１２に記憶されている解析情報から、プログラム構築部１３１において実行形式プログラムを生成しやすい形式のコードを生成するものである。また、ポストコードジェネレータ１３４は、プログラム構築部１３１において、構築された実行形式プログラムから、オフジェクトコードの実行形式プログラムを生成するものである。

【０１１４】

このため、解析情報記憶部１２に記憶されている解析情報が、例えば、特定のプログラミング言語（例えば、Ｊａｖａ）のバイトコードなどである場合には、プリコードジェネレータ１３３により、実行形式プログラムの生成がしやすい形式の例えば中間コードに変換され、これがプログラム構築部１３１に供給される。

【０１１５】

プログラム構築部１３１は、プリコードジェネレータ１３１から供給された例えば中間コードから、あるいは、プロファイル情報をも用いて、実行形式プログラムを構築する。この場合、構築された実行形式プログラムは、オブジェクトコードではない。そこで、ポストコードジェネレータ１３４は、プログラム構築部１３１において構築された例えば中間コードの実行形式プログラムから、オブジェクトコードの実行形式プログラムを生成し、これを実行形式プログラム記憶部２０に記憶する。

【０１１６】

このように、プリコードジェネレータ１３３、ポストコードジェネレータ１３４を実行形式生成部１３に備えることによって、ソースプログラム解析情報記憶部１２に生成された解析情報が、中間コードやＪａｖａのバイトコードなどの場合であっても、オブジェクトコードの実行形式プログラムを生成することができる。

【０１１７】

なお、プリコードジェネレータ１３３が、解析情報記憶部１２に記憶された例

えば中間コードなどの解析情報からからオブジェクトコードを生成することができるものであれば、ポストコードジェネレータ１３４を用いなくてもすむようにすることができる。また、プログラム構築部１３１が、例えば、Ｊａｖａのバイトコードなどから、実行形式プログラムを構築することができるものであれば、プログラム構築部１３１において生成された実行形式プログラムから、ポストコードジェネレータ１３４により、オブジェクトコードの実行形式プログラムを生成するようにすれば、プリコードジェネレータ１３１を用いなくてもすむようにすることができる。

【０１１８】

このような場合には、プリコードジェネレータ１３３のみを設け、中間コードなどから実行形式プログラムを構築する場合には、常に、オブジェクトコードに変換した後に、実行形式プログラムを生成するようにしたり、また、ポストコードジェネレータ１３４のみを設け、中間コードなどから実行形式プログラムを構築する場合には、常に、中間コードなどからそのまま実行形式プログラムを生成し、中間コードなどの実行形式プログラムをポストコードジェネレータ１３４によりオフジェクトコードに変換するようにしてもよい。

【０１１９】

また、プリコードジェネレータやポストコードジェネレータを備えた実行形式生成部と、これらを備えない実行形式生成部とを設けておくことにより、ソースプログラム解析情報記憶部１２に生成される解析情報が、図３に示したような中間コードであっても、あるいは、図５に示したようなアセンブリコードであっても、それらから実行形式プログラムを生成することができる。

【０１２０】

また、図９に示したように、実効形式生成部１３にプリコードジェネレータ１３３、ポストコードジェネレータ１３４を設けておき、アセンブリコードなどのオブジェクトコードが実行形式生成部１３に供給されたときには、プリコードジェネレータ１３３、ポストコードジェネレータ１３４をバイパスするようにしてもよい。

【０１２１】

［第２の実施の形態］

ところで、プロファイル情報は、前述の第１の実施の形態においても説明したように、実行形式プログラムを実行することにより得られる情報から生成される。しかし、コンパイル処理装置により生成した実行形式プログラムを、コンパイル処理装置において簡単に実行できない場合がある。

【０１２２】

例えば、生成した実行形式プログラムが、コンパイル処理を行った装置以外の計算機用の実行形式プログラムであり、ハードウエアの違いにより、プログラムの命令セットが違ったり、あるいは、ＯＳ（オペレーティングシステム）が異なるなど実行形式プログラムの実行環境が異なる場合である。このような場合、当該コンパイル処理装置においては、生成した実行形式プログラムを実行することができず、簡単にはプロファイル情報を得ることができない。

【０１２３】

また、目的とする実行形式プログラムが、例えば、ＶＴＲなどの電子機器の基板（ターゲットのボード）のメモリにインストールされて用いられるものである場合、例えば、テスト用の基板に実行形式プログラムをインストールして実行し、この実行により得られる情報をコンパイル処理装置に供給するようにしなければならない。

【０１２４】

しかし、この場合には、テスト用の基板を予め用意するとともに、テスト用の基板と、コンパイル処理装置とを多数のコード（ｃｏｒｄ）によって接続するようにしなければならない。また、この場合には、コンパイル処理装置にテスト用の基板上で目的とする実行形式のプログラムを実行してプロファイル情報を得るためのシュミレータを搭載しておくことも考えられるが、シュミレータの開発に時間やコストがかかる。

【０１２５】

そこで、この第２の実施の形態のコンパイル処理装置においては、プログラムの実行環境などに依存することなく実行が可能な、間接実行形式の中間コードプ

ログラムを生成し、これを最適化した実行形式プログラムを生成する当該コンパイル処理装置において実行することにより、簡単かつ迅速にプロファイル情報を得て、最適化した目的とする実行形式プログラムを迅速に生成するようにしたものである。

【０１２６】

図１０は、間接実行形式の中間コードプログラムを生成することができるコンパイル処理装置２を説明するための図である。図１０に示すように、この例のコンパイル処理装置２は、ソースプログラム解析部１１、ソースプログラム解析情報記憶部１２、実行形式生成部１３Ａ、１３Ｂ、１３Ｃ、プロファイル情報生成部１４、プロファイル情報記憶部１５、間接実行形式中間生成部１７を備えている。

【０１２７】

間接実行形式中間生成部１７を備えるとともに、中間コード、アセンブリコード、機械語などもそれぞれから実行形式プログラムを生成する実行形式生成部１３Ａ、１３Ｂ、１３Ｃを備える他は、図１を用いて前述した第１の実施の形態のコンパイル処理装置１と同様に構成されたものである。このため、図１０に示すこの第２の実施の形態のコンパイル処理装置２おいて、第１の実施の形態のコンパイル処理装置１と同様に構成される部分には、図１に示したコンパイル処理装置１と同じ参照符号を付し、その説明については省略する。

【０１２８】

なお、実行形式生成部１３Ａ、１３Ｂ、１３Ｃのそれぞれは、異なる機器で実行される直接実行形式プログラムを生成するために設けられたものである。このように、種類の異なる実行形式プログラムを生成するために複数の実行形式生成部１３Ａ、１３Ｂ、１３Ｃを設けるようにした場合であっても、プロファイル情報は、共通に用いることができる。

【０１２９】

また、この実施の形態においては、例えば、実行形式生成部１３Ａは中間コードを使用し、実行形式生成部１３Ｂは、アセンブリコードを使用し、実行形式生成部１３Ｃは、その他のオブジェクトコードを使用するというように、生成する

実行形式プログラムに応じて異なる解析情報を用いるようにされたものである。

【０１３０】

しかし、実行形式生成部１３Ａ、１３Ｂ、１３Ｃのそれぞれは、基本的には、前述した第１の実施の形態において、図２あるいは図９を用いて前述した実行形式生成部１３と同様に構成されたものであり、ソースプログラム解析情報記憶部１２からの解析情報から実行形式プログラムを生成することができるとともに、プロファイル情報記憶部１５のプロファイル情報から生成される確率情報をも用いることによって、最適化された実行形式プログラムを生成することができるものである。

【０１３１】

そして、この図１０に示すコンパイル処理装置２においては、間接実行形式中間生成部１７により、ソースプログラム解析情報記憶部１２に記憶されたソースプログラムの解析情報から、このコンパイル処理装置２において実行可能な間接実行形式の中間コードプログラムが生成され、これが中間プログラム記憶部３０に記憶される。

【０１３２】

この中間プログラム記憶部３０に記憶された間接実行形式の中間コードプログラムを実行することにより得られる情報から、プロファイル情報生成部１４によりプロファイル情報が生成されて、プロファイル情報記憶部１５に記憶される。

【０１３３】

そして、実行形式生成部１３Ａ、１３Ｂ、１３Ｃのいづれかにより、プロファイル情報記憶部１５に記憶されたプロファイル情報と、ソースプログラム解析情報記憶部１２に記憶されている解析情報とから、目的とする計算機において実行可能とされた最適化された間接実行形式プログラムが生成される。この第２の実施の形態において、実行形式生成部１３Ａ、１３Ｂ、１３Ｃのいづれが用いられるかは、例えば、このコンパイル処理装置２の操作者によって指示される。

【０１３４】

このように、最適化のために用いるプロファイル情報を生成するためにコンパイル処理装置により生成された実行形式プログラムが、当該コンパイル処理装置

においてすぐに実行できないような場合でも、計算機に依存することなく実行が可能な間接実行形式の中間コードプログラムを生成して、実行することにより、目的とするプログラムに対するプロファイル情報を得て、目的の計算機において実行可能とされた間接実行形式の実行形式プログラムを迅速に生成することができる。

【０１３５】

そして、この第２の実施の形態の場合には、それぞれの実行形式生成部１３Ａ、１３Ｂ、１３Ｃで生成された実行形式プログラムは、対応する実行形式プログラム２０Ａ、２０Ｂ、２０Ｃに記憶するようにされ、操作者の目的とする実行形式プログラムを生成することができる。

【０１３６】

なお、前述の実施の形態においては、実行形式生成部１３Ａ、１３Ｂ、１３Ｃのうちいづれを用いるかは、操作者が選択するものとして説明したがこれに限るものではない。例えば、中間コード、アセンブリコード、機械語などの使用する解析情報を操作者が選択することにより、実行形式生成部が自動的に決まるようにすることもできる。つまり、使用する解析情報と、使用する実行形式生成部とを１対１に対応させるようにコンパイル処理装置２を構成することもできる。

【０１３７】

また、使用する解析情報と、使用する実行形式生成部との両方を操作者が選択して、目的とする実行形式プログラムを生成するようにコンパイル処理装置２を構成するようにすることもできる。

【０１３８】

また、複数の実行形式生成部を設けるのではなく、例えば、操作者からの指示に応じて、供給される解析情報が、中間コード、アセンブリコード、機械語、その他のオブジェクトコードなどであっても、それらに応じて実行形式プログラムを生成することができる多機能の実行形式生成部を設けるようにすることもできる。

【０１３９】

このように、この第２の実施の形態のコンパイル処理装置２は、例えば、操作

者からの指示に応じて、操作者が目的とする実行形式プログラムを生成することができるものである。

【０１４０】

また、実行形式生成部に供給される解析情報が、中間コード、アセンブリコード、機械語などを使い分けることができるのと同様にして、中間実行形式中間生成部１７にも、中間コードを供給するか、アセンブリコードを供給するか、機械語を供給するかなどを切り換えることができるようにすることもできる。

【０１４１】

したがって、中間実行形式中間生成部１７に供給される解析情報と、実行形式生成部１３Ａ、１３Ｂ、１３Ｃに供給される解析情報とが、異なる種類のコードとなる場合もあるし、同じ場合になることもある。要するに、種々の解析情報の使い分けをできるようにしておくことにより、より柔軟に各種の間接実行形式プログラム（中間コードプログラム）や直接実行形式プログラムを生成することができるようにされる。

【０１４２】

また、間接実行形式中間生成部１７により中間コードプログラムを生成するのか、実行形式生成部１３Ａ、１３Ｂ、１３Ｃのいづれかにより実行形式プログラムを生成するのかなどの切り換えは、例えば、コンパイル処理装置１に設けられた切り換えスイッチなどを操作することにより切り換えることができるようにされる。

【０１４３】

［第３の実施の形態］

例えば、１つのコンパイル処理装置により直接実行形式と、間接実行形式とのいずれの実行形式プログラムをも生成可能にする場合や、プログラムの実行環境が異なる種々の計算機で実行される実行形式プログラムを生成しようとする場合には、直接実行形式と間接実行形式との別や、目的とする計算機のプログラムの実行環境などに関する情報などをコンパイル処理時にコンパイル処理装置に入力する必要が生じる。

【０１４４】

そこで、この第３の実施の形態のコンパイル処理装置は、コンパイル処理装置の操作者からのコンパイル指示情報や、コンパイル処理を補助するためのプログラムから提供されるコンパイル指示情報を受け付けて、受け付けたコンパイル指示情報をも考慮して実行形式プログラムを生成することができるようにしたものである。

【０１４５】

図１１は、この第３の実施の形態のコンパイル処理装置３を説明するための図である。図１１に示すように、この第３の実施の形態のコンパイル処理装置は、ソースプログラム解析部１１、ソースプログラム解析情報記憶部１２、実行形式生成部１３、プロファイル情報生成部１４、プロファイル情報記憶部１５、指示情報受付部１８を備えている。

【０１４６】

指示情報受付部１８以外の各部は、図１を用いて前述した第１の実施の形態のコンパイル処理装置１の各部と同様に構成されたものである。このため、図１１に示すこの第３の実施の形態のコンパイル処理装置３おいて、第１の実施の形態のコンパイル処理装置１と同様に構成される部分には、図１に示したコンパイル処理装置１と同じ参照符号を付し、その説明については省略する。

【０１４７】

また、この例においても、ソースプログラム格納部１０には、図３に示したソースプログラムが格納されており、このソースプログラムをソースプログラム解析部１１が解析することにより生成した、例えば、図５に示したアセンブリコードが、ソースプログラム解析情報記憶部１２に記憶されているものとして説明する。

【０１４８】

指示情報受付部１８は、操作者３１から、例えば、キーボード装置やいわゆるマウスなどのポインティングデバイス、あるいは、コンパイルを補助するプログラムである例えばグラフィカルユーザインターフェース（ＧＵＩ：Ｇｒａｐｈｉｃａｌ　Ｕｓｅｒ　Ｉｎｔｅｒｆａｃｅ）３２を通じて入力された種々のコンパ

イル指示情報を受け付けて、これを実行形式生成部１３に供給する。

【０１４９】

ここで、コンパイル指示情報は、例えば、間接実行形式と、直接実行形式の別などの情報や、コンパイル処理を制御するための情報などである。コンパイル処理を制御するための情報としては、例えば、コードの配置を制御するための情報や、ある特定のブロックを最適化するように指示するなどの最適化を制御するための情報などがある。

【０１５０】

また、指示情報受け付け部１８は、目的とする実行形式プログラムを生成するためにコンパイル処理において必要となるパラメータや、コンパイル処理に必要となるデータが蓄積されたファイルの指示情報などを与えるコンパイル処理を補助するためのプログラムからのコンパイル指示情報を受け付けて、これを実行形式生成部１３に供給する。

【０１５１】

そして、実行形式生成部１３は、指示情報受付部１８からのコンパイル指示情報をも考慮して、ソースプログラム解析情報記憶部１２の解析情報から、または、ソースプログラム解析情報記憶部１２の解析情報と、プロファイル情報記憶部１５のプロファイル情報とから、コンパイル指示情報に基づいた実行形式プログラムを生成する。

【０１５２】

これにより、操作者やコンパイル処理を補助するためのプログラムからのコンパイル指示情報をも考慮して、最適化された目的とする実行形式プログラムを簡単かつ迅速に生成することができる。

【０１５３】

次のこの第３の実施の形態のコンパイル処理装置について、具体例を用いてより詳細に説明する。

この第３の実施の形態のコンパイル処理装置３においては、ソースプログラム解析部１１においての解析結果や、実行形式生成部１３において生成された実行形式プログラムの生成結果などの情報、あるいは、プロファイル情報を統計処理

することにより形成される確率情報などの各種の情報が、この実施の形態のコンパイル処理装置３に接続されたディスプレイに表示されるなどして操作者に通知することができるようにされている。

【０１５４】

操作者は、ディスプレイに表示された情報を確認しながら各種のコンパイル指示情報を入力することができるようにされる。すなわち、この第３の実施の形態においては、グラフィッカルユーザインターフェース３２が実現され、このグラフィカルユーザインターフェースを通じてコンパイル指示情報をコンパイル処理装置３に入力することができるようにされている。

【０１５５】

そして、操作者は、例えばディスプレイに表示されるなどして通知された情報などに基づいて、キーボードやマウスなどのポインティングデバイスや、グラフィカルユーザインターフェースを通じて、コンパイル指示情報を入力する。入力されたコンパイル指示情報は、指示情報受付部１８を通じて、実行形式生成部１３に供給される。

【０１５６】

図１２は、この第３の実施の形態のコンパイル処理装置３の指示情報受付部１８と、この指示情報受付部１８からのコンパイル指示情報の供給を受ける実行形式生成部１３とを説明するための図である。例えば、コンパイル指示情報が、間接実行形式と直接実行形式の別を指示する情報である場合には、受け付けられたコンパイル指示情報は、実行形式生成部１３のプログラム構築部１３１に供給される。

【０１５７】

また、受け付けられたコンパイル指示情報が、プロファイル情報から生成された確率情報を変更することによって、目的とする最適化を行うようにする最適化を制御するための情報である場合には、受け付けられた情報は、確率情報生成部１３２に供給される。

【０１５８】

図１３は、グラフィカルユーザインターフェース３２の表示例を示す図である

。この例は、前述したように、確率情報生成部１３２により、プロファイル情報から生成された確率情報（図７）のうち、分岐の確率を示すとともに、分岐の確率の変更を行うための画面を示している。

【０１５９】

前述もしたように、図５に示したアセンブリコードの場合を例にとると、ブロック２を実行する確率（変数ａが変数ｂよりも大きい確率）は１０％であり、図１３においてＴ＝１０％のように表示されている。また、ブロック３が実行される確率（変数ｂが変数ａ以上である確率）は９０％であり、図１３においてＦ＝９０％のように表示されている。

【０１６０】

この分岐の確率を変更して最適化を行いたい場合には、図１３において矢印が示すように、最適化を制御するための情報の変更入力領域において、確率を変更する。図１３の例の場合には、分岐の確率を逆にし、ブロック２を実行する確率（変数ａが変数ｂよりも大きい確率）を９０％に変更し、ブロック３が実行される確率（変数ｂが変数ａ以上である確率）を１０％に変更するようにしている。

【０１６１】

この例の場合の最適化を制御するための情報は、図１２を用いて前述したように、実行形式生成部１３の確率情報生成部１３２に供給され、分岐の確率情報が変更するようにされる。そして、実行形式生成部１３においては、指示情報受付部１８から供給されたコンパイル処理を補助するためのプログラム、例えば、グラフィッカルユーザインターフェースからのコンパイル指示情報をも考慮して、前述したように、実行形式プログラムを生成する。

【０１６２】

図１４は、図１３を用いて前述したように、分岐の確率情報が変更されて最適化が行われて生成された実行形式プログラムをアセンブリコードで表したものである。図５に示したアセンブリコードが最適化前のアセンブリコードであり、分岐の確率が変更されることにより、図１４に示すように、ｂｌｏｃｋ１→ｂｌｏｃｋ２→ｂｌｏｃｋ３の経路が最適化される。

【０１６３】

この図１４に示す最適化されたアセンブリコードと、図５に示した最適化される前のアセンブリコードとを比較すると分かるように、確率情報を用いて最適化を行うと、最適化後のアセンブリコードのｂｌｏｃｋ１の３行目には、新たにレジスタｒ１の値（変数ａ）をレジスタｒ４に格納するコードが追加され、ｂｌｏｃｋ２のコードと、ｂｌｏｃｋ４のコードが統合するようにされている。

【０１６４】

これにより、ｂｌｏｃｋ１において、レジスタｒ１の値（変数ａ）がレジスタｒ４に格納されているので、変数ａが変数ｂより大きいときには、ｂｌｏｃｋ１に続くｂｌｏｃｋ２において、変数ａの２乗が即座に求められ変数ｄに格納することができるようにされる。すなわち、２つのｂｌｏｃｋの処理で、変数２の２乗を求めることができるようにされる。

【０１６５】

また、変数ｂが変数ａより大きい場合には、図５に示した最適化前のアセンブリコードの場合と同様に、ｂｌｏｃｋ３において変数ｂが変数ｃに格納され、ｂｌｏｃｋ４において、変数ｃ（すなわち変数ａ）が２乗されて変数ｄに格納されるというように、３つのブロックの処理で、変数ａの２乗を求めることができるようにされる。

【０１６６】

したがって、発生する確率が９０％とされた変数ａが変数ｂよりも大きい場合の処理が、迅速に行われるように、最適化された実行形式プログラムが生成される。なお、図１４においては示さなかったが、ｂｌｏｃｋ２からジャンプする先のｂｌｏｃｋ５は、ｂｌｏｃｋ４に続くブロックである。

【０１６７】

このように、この第３の実施の形態のコンパイル処理装置３は、操作者やコンパイル処理を補助するためのプログラムからのコンパイル指示情報をも考慮して、最適化された目的とする実行形式プログラムを簡単かつ迅速に生成することができる。つまり、操作者は、手動でコンパイル指示情報をコンパイル処理装置３に対して与えることができるので、コンパイル処理装置が自動ではできない細か

い制御を行うことが可能になり、より効率のよい実行形式プログラムを生成することができる。

【０１６８】

また、例えば、操作者は、プロファイル情報を得るためにソースプログラム解析情報記憶部１２に記憶されている解析情報に基づいた実行形式プログラムを作成するのか、プロファイル情報を考慮して、最適化した実行形式プログラムを生成するのかなどの指示も、指示情報受付部１８を通じて、実行形式生成部１３に供給することができる。

【０１６９】

［第４の実施の形態］

前述した第３の実施の形態においては、コンパイル処理の実行時にコンパイル指示情報を指示情報受付部１８を通じて、実行形式生成部１３に供給するようにした。しかし、通常コンパイル指示情報は、そのほとんどがコンパイル処理前には分かっている場合が多い。

【０１７０】

このため、コンパイル処理前に分かっているコンパイル指示情報については、予めコンパイル処理装置に与えておき、コンパイル処理装置においてのコンパイル処理を迅速に行うようにするとともに、例えば、コンパイル指示情報が間違っていた場合などにおいては、間違っていたコンパイル指示情報を修整するだけで、すなわち、再度、すべてのコンパイル指示情報の入力をすることなくコンパイル処理を行って、実行形式プログラムを迅速に生成することができるようにしておくことが望ましい。

【０１７１】

そこで、この第４の実施の形態のコンパイル処理装置は、操作者やコンパイル処理を補助するためのプログラムからのコンパイル指示情報を記憶保持するコンパイル指示情報記憶部を設け、コンパイル処理を実行する都度、コンパイル指示情報の入力をすることがないようにしたものである。

【０１７２】

図１５は、この第４の実施の形態のコンパイル処理装置４を説明するための図

である。図１５に示すように、この第４の実施の形態のコンパイル処理装置は、コンパイル指示情報記憶部１９以外の各部は、図１１を用いて前述した第３の実施の形態のコンパイル処理装置３の各部と同様に構成されたものであり、指示情報受付部１８、コンパイル指示情報記憶部１９を除けば、第１の実施の形態のコンパイル処理装置１と同様に構成されたものである。

【０１７３】

そして、この第４の実施の形態のコンパイル処理装置４において、指示情報受付手段１８は、操作者やコンパイル処理を補助するためのプログラムからのコンパイル指示情報を受け付けて、これをコンパイル指示情報記憶部１９に記憶する。

【０１７４】

コンパイル指示情報記憶部１９に記憶されたコンパイル指示情報は、実行形式プログラムの生成時に、実行形式生成部１３により読み出される。そして、実行形式生成部１３は、コンパイル指示情報記憶部１９から読み出したコンパイル指示情報をも考慮して、実行形式プログラムを生成する。

【０１７５】

これにより、操作者やコンパイル処理を補助するためのプログラムからのコンパイル指示情報を予めコンパイル指示情報記憶部１９に記憶させておくことにより、コンパイル処理実行時において、予め決まっていた数多くのコンパイル指示情報を入力したり、コンパイル処理を補助するためのプログラムを実行しなくても済むので、より迅速に、目的とする計算機で実行可能とされた目的の実行形式の実行形式プログラムを生成することができる。

【０１７６】

つまり、コンパイル処理時においてコンパイル処理装置に提供するコンパイル指示情報を必要最小限に押さえることができるなど、コンパイル指示情報の入力にかかる時間を短縮することができ、目的とする実行形式プログラムを迅速に生成することができる。

【０１７７】

なお、前述の実施の形態において示したソースプログラムや中間コード、アセ

ンブリコード、プロファイル情報、確率情報などは一例であり、この発明のコンパイル処理装置を用いることによって、各種のソースプログラムからそのソースプログラムに応じた解析情報の生成、実行形式プログラムの生成、プロファイル情報の生成、最適化した実行形式プログラムの生成を行うことができる。

【０１７８】

また、コンパイル指示情報は、前述した実行形式の別や、目的とする計算機のプログラムの実行環境に関する情報、コンパイルを制御する情報に限るものではなく、目的の実行形式プログラムを生成するためにコンパイル処理において必要となる種々の情報をも含むものである。

【０１７９】

また、前述の第３、第４の実施の形態のコンパイル処理装置３、４に、前述した第２の実施の形態の間接実行形式中間生成部１７を搭載することもできる。この場合には、操作者などからの中間コードプログラムを生成する指示を間接実行形式中間生成部１７に供給して、中間コードプログラムを生成するようにすることができる。

【０１８０】

また、前述した実施の形態のコンパイル処理装置は、ソフトウエアによって構成するようにすることができる。すなわち、前述した各実施の形態のコンパイル処理装置の機能を備えた、コンパイル処理プログラムを形成することができる。

【０１８１】

また、前述した実施の形態のコンパイル処理装置の実行形式生成部は、関連する複数の実行形式プログラムを相互にリンクさせて、１つの実行形式モジュールを生成するようにすることもできるものである。

【０１８２】

また、前述した各実施の形態のコンパイル処理装置は、単独で使用するようにすることもできるが、種々の電子計算機に搭載して利用するようにすることができる。

【０１８３】

また、前述した各実施の形態のコンパイル処理装置は、扱うプログラムのデー

タ構造などに依存することなく、目的とする計算機において実行可能な実行形式プログラムを生成することができる。

【０１８４】

すなわち、前述した各実施の形態のコンパイル処理装置は、これが搭載される計算機のハードウエア構成や、扱うプログラムなどのデータのデータ構造などに左右されることなく、目的とする計算機などの電子機器において実行可能な実行形式プログラムを生成することができるものである。

【０１８５】

【発明の効果】

以上説明したように、請求項１に記載の発明のコンパイル処理装置によれば、最適化された実行形式プログラムを生成する場合、ソースプログラムを繰り返し用いることなく、ソースプログラムを解析処理することにより生成されたソースプログラムの解析情報とプロファイル情報とに基づいて、最適化された実行形式のプログラムを生成することができる。したがって、最適化された実行形式のプログラムを生成する場合に、プロファイル情報を生成するための実行形式のプログラムを生成するために行ったソースプログラムについての解析処理を重複して行う必要がなくなり、迅速に最適化した実行形式プログラムを生成することができる。

【０１８６】

また、請求項２に記載の発明のコンパイル処理装置によれば、プロファイル情報を得るために実行するプロファイル情報生成用の最適化前の実行形式プログラムも、最適化された実行形式プログラムも、同じ実行形式プログラム生成部により生成することができる。したがって、プロファイル情報生成用実行形式プログラム生成部と、最適化実行形式プログラム生成部とを別々に設ける必要がなくなるり、コンパイル処理装置の構成を簡単にすることができる。

【０１８７】

また、請求項３に記載の発明のコンパイル処理装置によれば、目的とする計算機において実行可能であるその計算機に固有な形式であって、その目的とする計算機のプログラムの実行環境などに適合したプログラムを生成することができる

。また、この直接実行形式のプログラムをコンパイル処理装置を有する装置において実行することにより、その直接実行形式のプログラムのプロファイル情報を得て、最適化された実行形式のプログラムを生成することができる。

【０１８８】

また、請求項４に記載の発明のコンパイル処理装置によれば、間接実行形式のプログラムが生成できる。間接実行形式のプログラムは、これを実行する計算機などの機器に依存することなく実行することができるので、コンパイル処理装置において、間接実行形式のプログラムを実行することにより、プロファイル情報を得て、このプロファイル情報をも用いることによって、最適化した実行形式のプログラムを作成することができる。

【０１８９】

また、請求項５に記載の発明のコンパイル処理装置によれば、プログラムが実行される計算機のプログラムの実行環境などに依存することなく、間接実行形式中間プログラム生成部により生成されるいわゆる中間コードプログラムを実行することにより、そのプログラムに対するプロファイル情報を得て、このプロファイル情報を用いることにより、目的とする間接実行形式の最適化されたプログラムを簡単かつ迅速に生成することができる。

【０１９０】

また、請求項６に記載の発明のコンパイル処理装置によれば、操作者からのコンパイル指示情報をも考慮した実行形式プログラムを簡単かつ迅速に生成することができる。

【０１９１】

また、請求項７に記載の発明のコンパイル処理装置によれば、操作者は、コンパイル処理前に予め決まっているコンパイル指示情報について、予めコンパイル指示情報記憶部に記憶させておくことができる。したがって、コンパイル処理実行時のコンパイル指示情報の入力処理にかかる時間を短縮することができ、コンパイル指示情報記憶部に記憶されたコンパイル指示情報をも考慮した、操作者からの要求に応じた実行形式プログラムをより迅速に生成することができる。

【０１９２】

また、請求項８に記載の発明のコンパイル処理装置によれば、コンパイル処理を補助するためのプログラムからのコンパイル指示情報をも考慮して、実行形式プログラムが生成されるので、目的とする実行形式プログラムを簡単かつ迅速に生成することができる。

【０１９３】

また、請求項９に記載の発明のコンパイル処理装置によれば、コンパイル処理に必要なコンパイル処理を補助するプログラムからの種々のコンパイル指示情報を予めコンパイル指示情報記憶部に記憶させておくことができるので、コンパイル処理時にコンパイル処理を補助するためのプログラムを実行させる必要がなくなる。これにより、コンパイル指示情報記憶部に記憶されたコンパイル指示情報をも考慮して、目的とする実行形式プログラムをより迅速に生成することができる。

【図面の簡単な説明】

【図１】

この発明によるコンパイル処理装置の一実施の形態を説明するための図である。

【図２】

図１に示した実行形式生成部１３を説明するための図である。

【図３】

ソースプログラムの一例を示す図である。

【図４】

ソースプログラム解析部により生成される解析情報としての中間コードの一例を説明するための図である。

【図５】

ソースプログラム解析部により生成される解析情報としてのアセンブリコードの一例を説明するための図である。

【図６】

プロファイル情報生成部により生成されるプロファイル情報の一例を説明する

ための図である。

【図７】

実行形式生成部の確率情報生成部において生成される確率情報の一例を説明するための図である。

【図８】

プロファイル情報から生成される確率情報が用いられて最適化されたプログラムの例を説明するための図であるい。

【図９】

コンパイル処理装置の実行形式生成部の他の例を説明するための図である。

【図１０】

この発明によるコンパイル処理装置の一実施の形態の他の例を説明するための図である。

【図１１】

この発明によるコンパイル処理装置の一実施の形態の他の例を説明するための図である。

【図１２】

図１１に示したコンパイル処理装置の指示情報受付部と実行形式生成部とを説明するための図である。

【図１３】

コンパイルを補助するためのプログラムとしてのグラフィッカルユーザインターフェースの一例を説明するための図である。

【図１４】

操作者からのコンパイル指示情報をも考慮して最適化されたプログラムの一例を説明するための図である。

【図１５】

この発明によるコンパイル処理装置の一実施の形態の他の例を説明するための図である。

【図１６】

従来のコンパイル処理装置を用いて最適化された実行形式プログラムを生成す

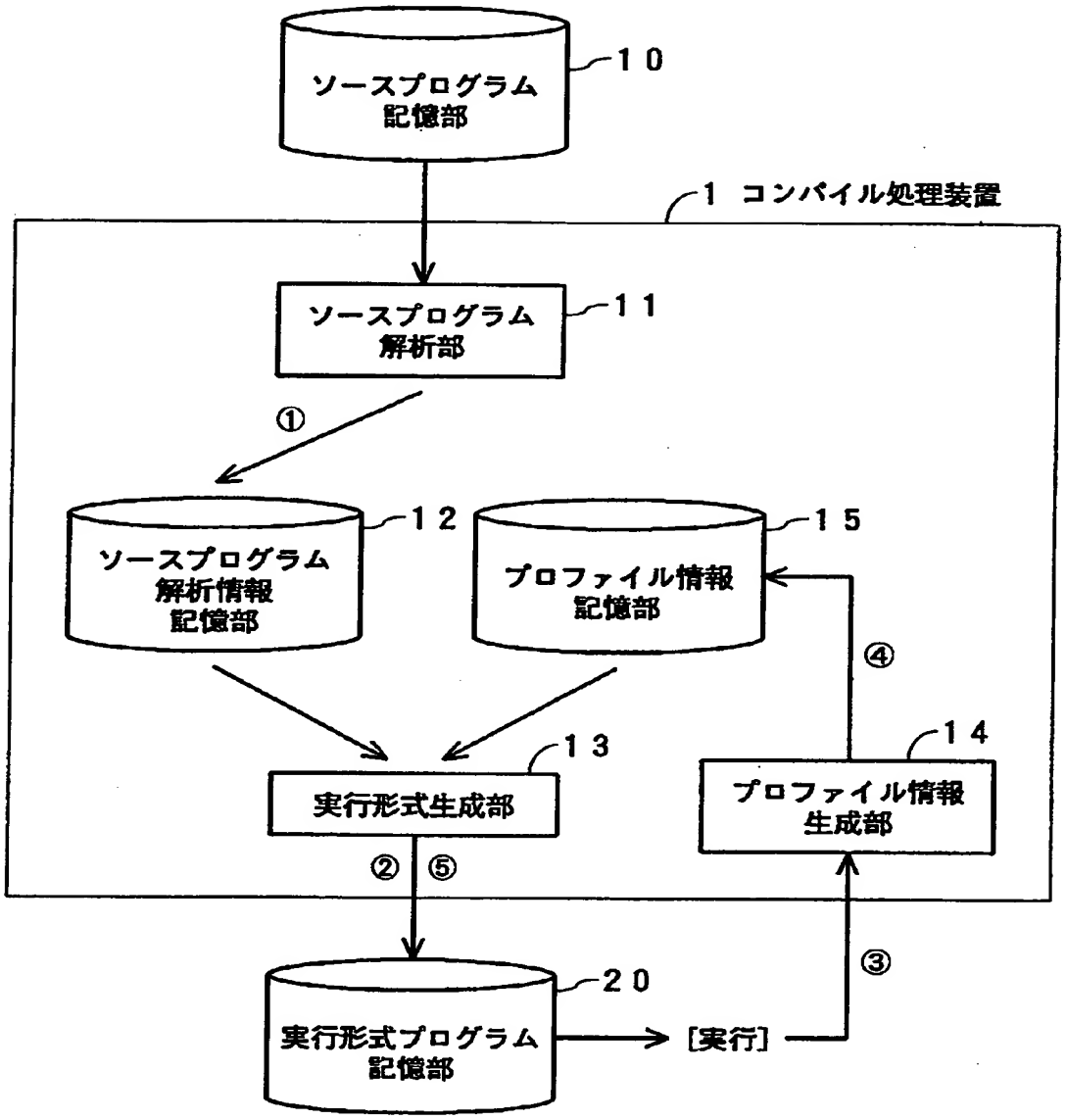
る場合の手順を説明するための図である。

【符号の説明】

１、２…コンパイル処理装置、３、４…コンパイル処理装置、１１…ソースプログラム解析部、１２…ソースプログラム解析情報記憶部、１３…実行形式生成部、１３１…プログラム構築部、１３２…プログラム実行の確率情報生成部、１３３…プリコードジェネレータ（ｐｒｅ－ｃｏｄｅｇｅｎｅｒａｔｏｒ）、１３４…ポストコードジェネレータ（ｐｏｓｔ－ｃｏｄｅｇｅｎｅｒａｔｏｒ）、１４…プロファイル情報生成部、１５…プロファイル情報記憶部、１７…間接実行形式中間生成部、１８…指示情報受付部、１９…コンパイル指示情報記憶部、１０…ソースプログラム記憶部、２０…実行形式プログラム記憶部、３０…中間プログラム記憶部

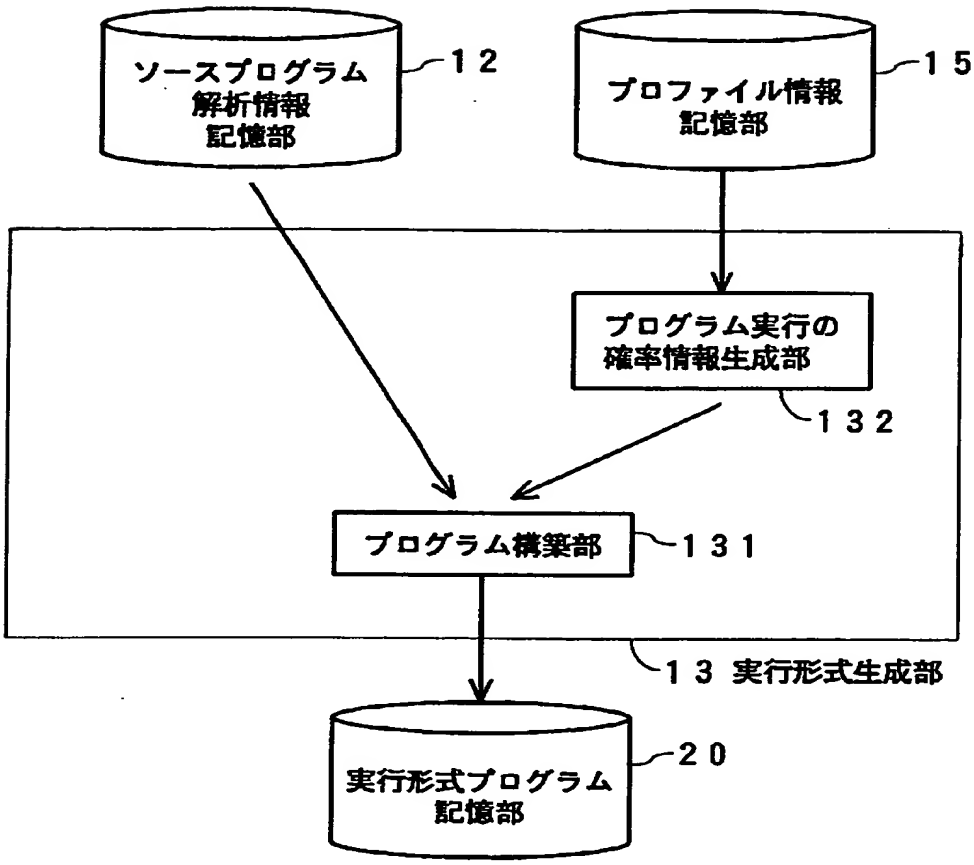
【書類名】　図面

【図１】



出証特平１１－３０５３２７４

【図２】


ソースプログラム
解析情報
記憶部
１２
プロファイル情報
記憶部
１５
プログラム実行の
確率情報生成部
１３２
プログラム構築部
１３１
１３ 実行形式生成部
実行形式プログラム
記憶部
２０

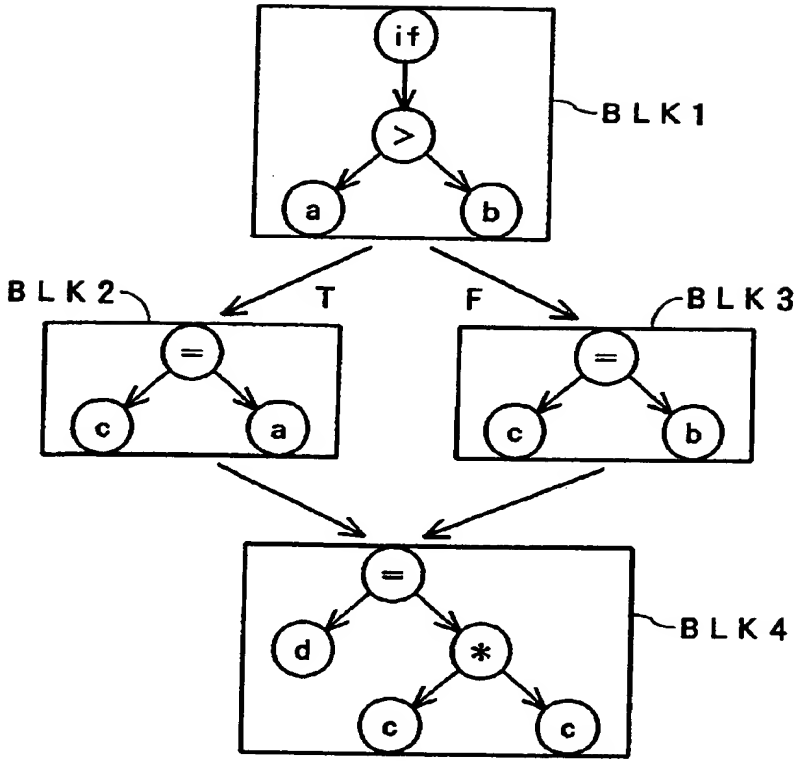
【図３】

ソースプログラムの例

```
if( a > b )      then  c = a ;
                 else  c = b ;
d = c * c ;
```

【図４】

中間コードの例



【図５】

アセンブリコード

```
.text
block_1:
        ld      r1, [a]             ; r1 <- a
        ld      r2, [b]             ; r2 <- b
        cmp     r3, r1, r2          ; r3 <- r1 cmp r2
        ble     r3, block_3
block_2:
        ld      r4, [a]             ; r4 <- a
        st      r4, [c]             ; c <- r4
        jmp     block_4
block_3:
        ld      r5, [b]             ; r5 <- b
        st      r5, [c]             ; c <- r5
        jmp     block_4
block_4:
        ld      r6, [c]             ; r6 <- c
        ld      r7, [c]             ; r7 <- c
        mul     r8, r6, r7          ; r8 <- r6 * r7
        st      r8, [d]             ; d <- r8
```

【図６】

プロファイル情報

経過時間	詳細情報	
...		
10050:	PC	block_1
	load address	a
10051:	load address	b
10052:	compare	
10053:	branch less or equal	block_3
10054:	PC	block_3
	load address	b
10055:	store address	c
10056:	jump	block_4
10057:	PC	block_4
	load address	c
10058:	load address	c
10059:	mul	
10060:	store address	d
...		

【図７】

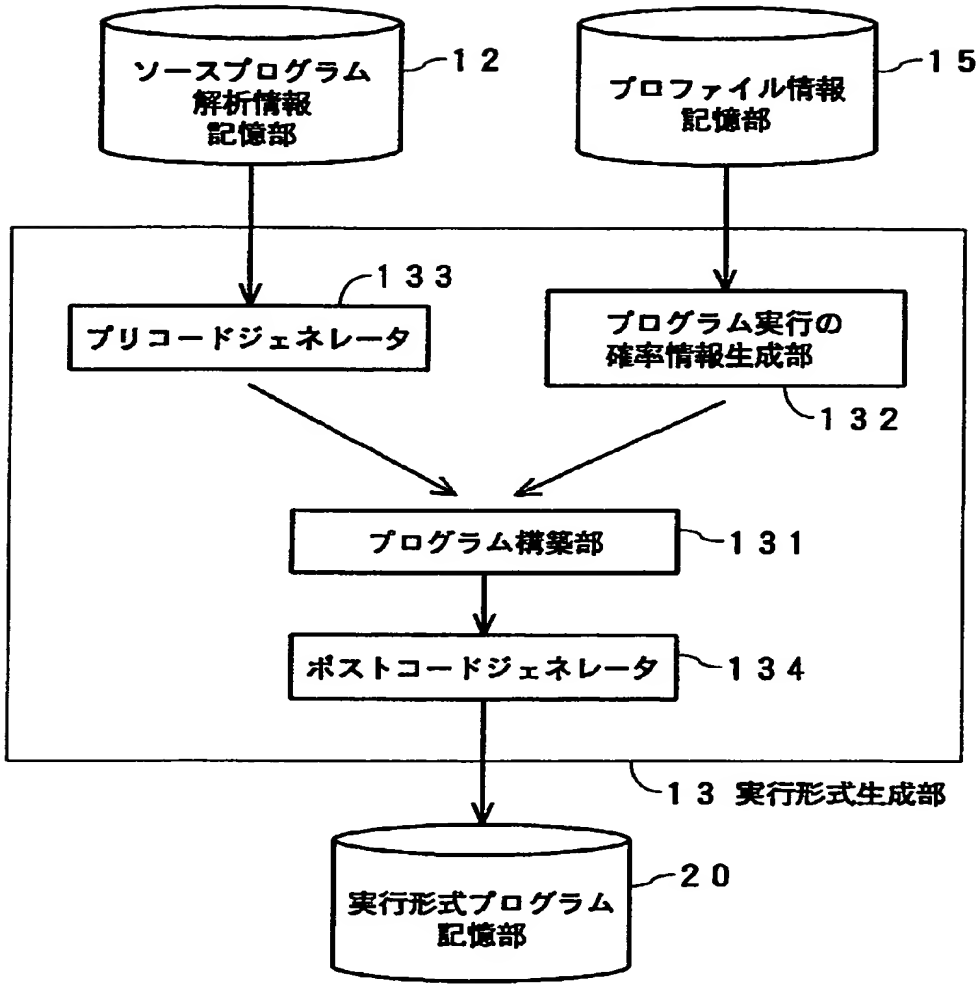
確率情報

ラベル名	実行/アクセス回数
...	
block_1:	100
block_2:	10
block_3:	90
block_4:	100
...	
a load:	110
a store:	0
b load:	190
b store:	0
c load:	200
c store:	100
d load:	0
d store:	100
...	

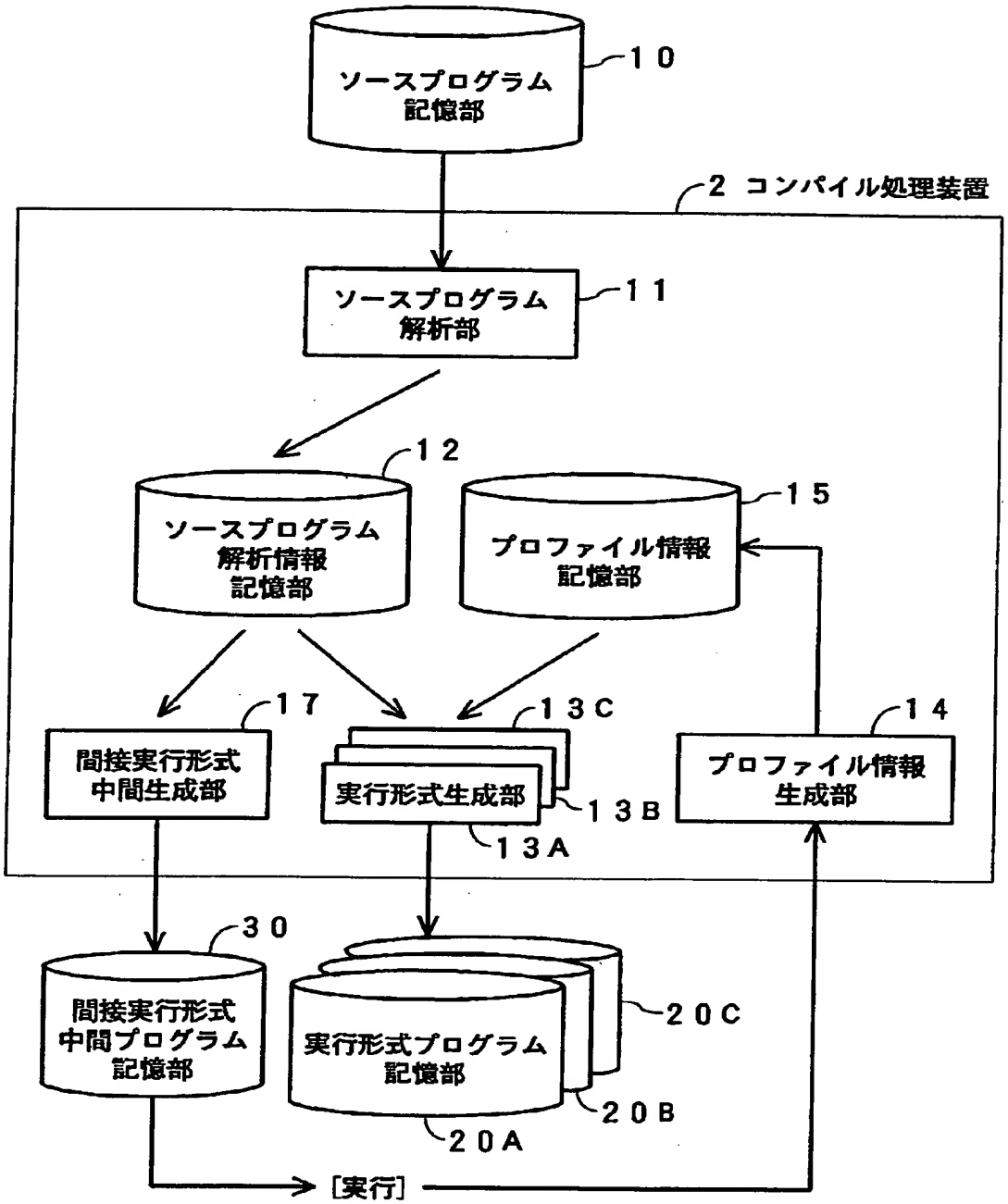
【図８】

```
.text
block_1:
        ld      r1, [a]             ; r1 <- a
        ld      r2, [b]             ; r2 <- b
        mv      r5, r2              ; r5 <- r2
        cmp     r3, r1, r2          ; r3 <- r1 cmp r2
        ble     r3, block_3
block_2:
        ld      r4, [a]             ; r4 <- a
        st      r4, [c]             ; c <- r4
        jmp     block_4
block_3:
        st      r5, [c]             ; c <- r5
        mul     r8, r5, r5          ; r8 <- r5 * r5
        st      r8, [d]             ; d <- r8
        jmp     block_5
block_4:
        ld      r6, [c]             ; r6 <- c
        ld      r7, [c]             ; r7 <- c
        mul     r8, r6, r7          ; r8 <- r6 * r7
        st      r8, [d]             ; d <- r8
```

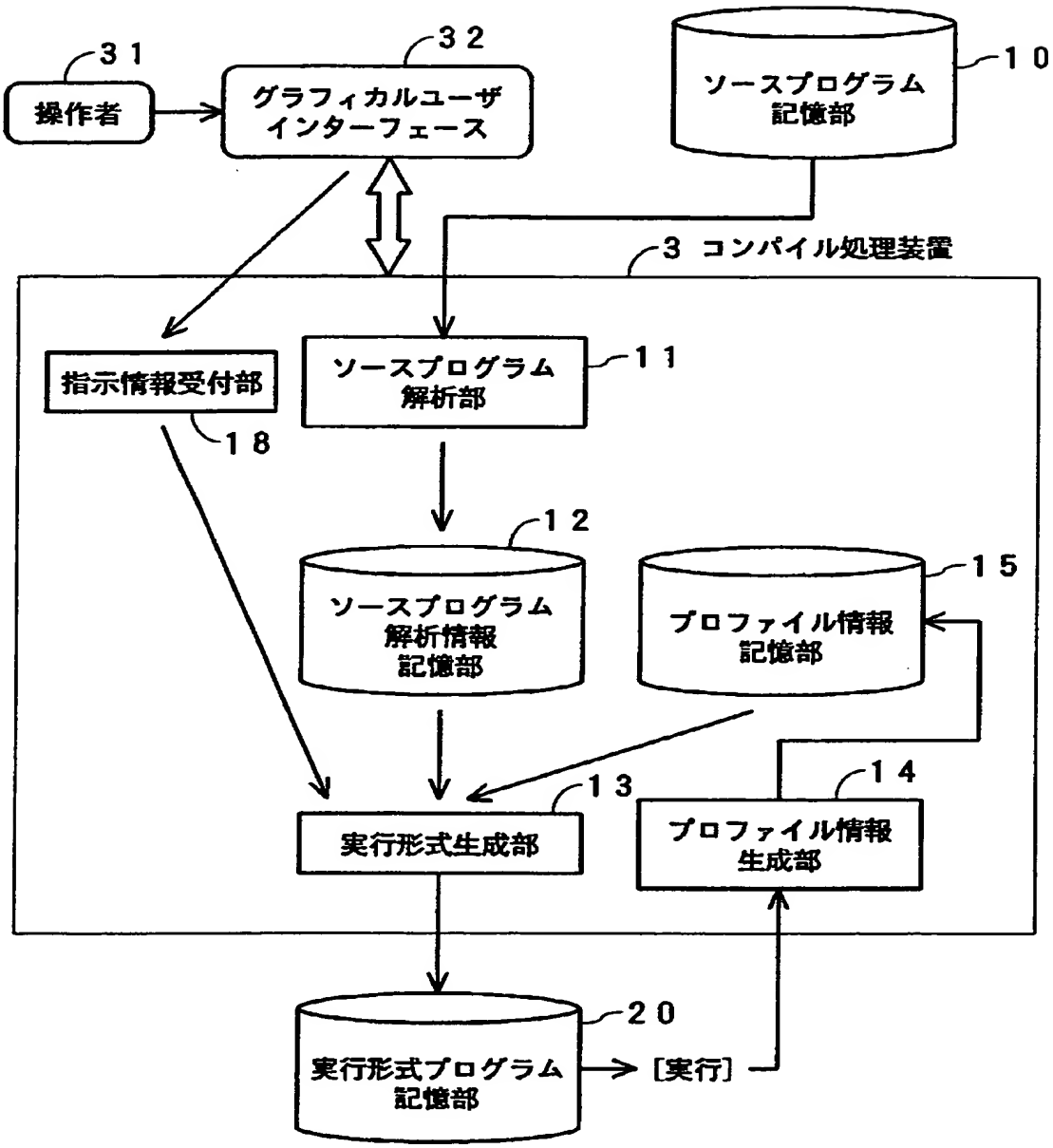
【図９】


ソースプログラム
解析情報
記憶部
１２
プロファイル情報
記憶部
１５
１３３
プリコードジェネレータ
プログラム実行の
確率情報生成部
１３２
プログラム構築部
１３１
ポストコードジェネレータ
１３４
１３ 実行形式生成部
実行形式プログラム
記憶部
２０

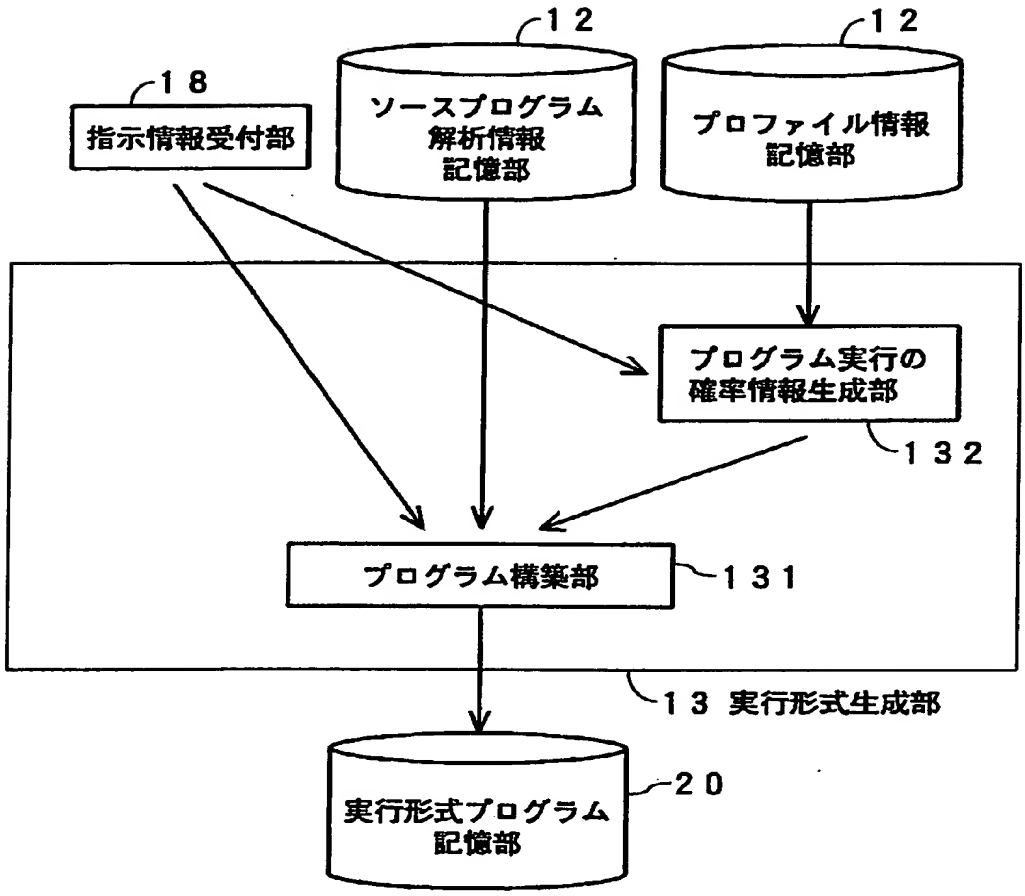
【図１０】


ソースプログラム
記憶部
１０
２ コンパイル処理装置
ソースプログラム
解析部
１１
１２
ソースプログラム
解析情報
記憶部
１５
プロファイル情報
記憶部
１７
間接実行形式
中間生成部
１３Ｃ
実行形式生成部
１３Ｂ
１３Ａ
１４
プロファイル情報
生成部
３０
間接実行形式
中間プログラム
記憶部
２０Ｃ
実行形式プログラム
記憶部
２０Ｂ
２０Ａ
［実行］

【図１１】

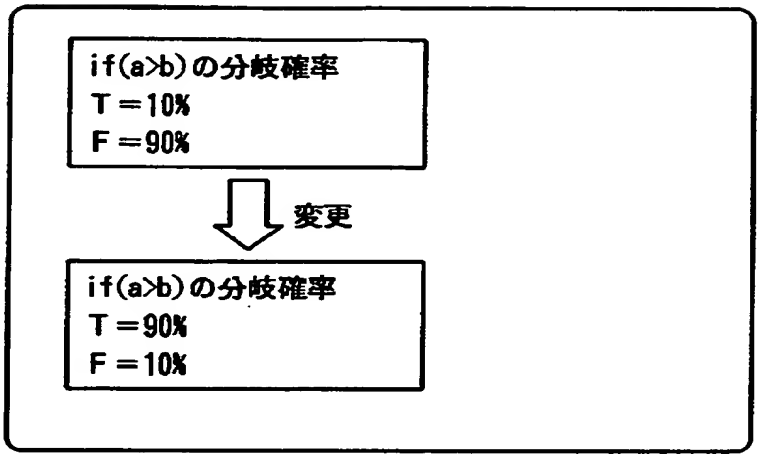

３１
操作者
３２
グラフィカルユーザ
インターフェース
ソースプログラム
記憶部
１０
３ コンパイル処理装置
指示情報受付部
１８
ソースプログラム
解析部
１１
１２
ソースプログラム
解析情報
記憶部
１５
プロファイル情報
記憶部
１３
実行形式生成部
１４
プロファイル情報
生成部
２０
実行形式プログラム
記憶部
[実行]

【図１２】


１２
１２
１８
指示情報受付部
ソースプログラム
解析情報
記憶部
プロファイル情報
記憶部
プログラム実行の
確率情報生成部
１３２
プログラム構築部
１３１
１３ 実行形式生成部
実行形式プログラム
記憶部
２０

【図１３】

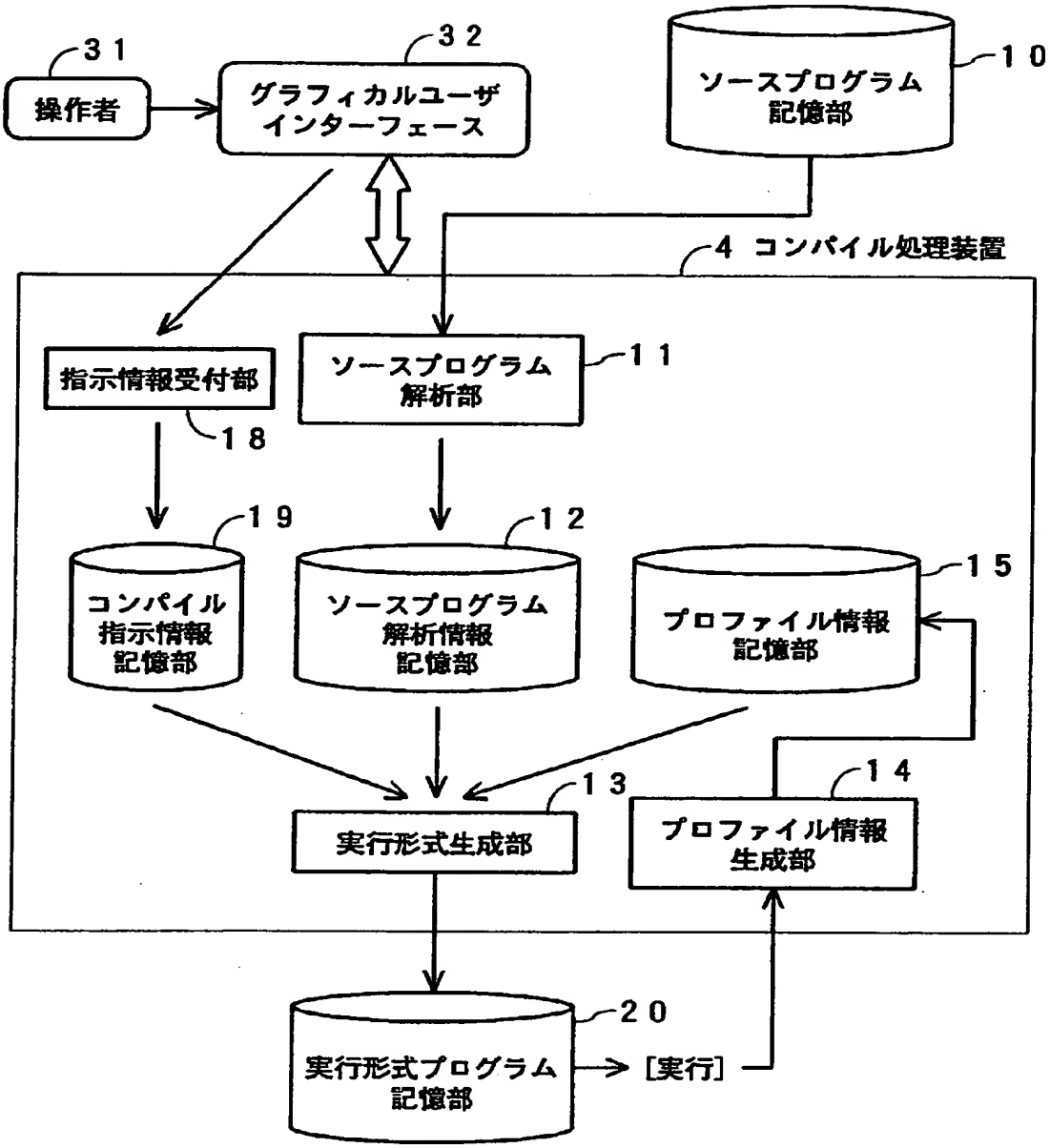
コンパイルを補助するためのプログラム



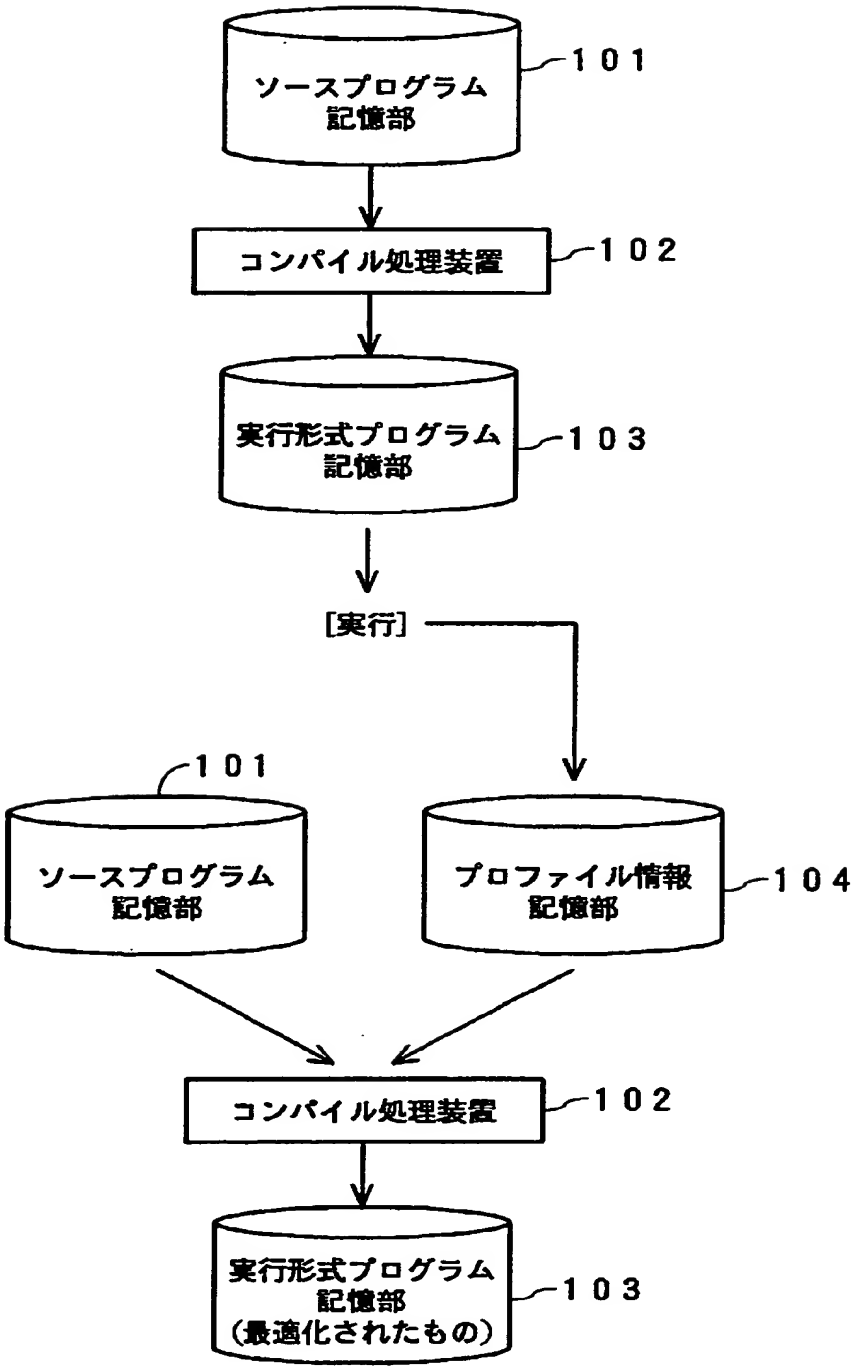
【図１４】

```
.text
block_1:
        ld      r1, [a]             ; r1 <- a
        ld      r2, [b]             ; r2 <- b
        mv      r4, r1              ; r4 <- r1
        cmp     r3, r1, r2          ; r3 <- r1 cmp r2
        ble     r3, block_3
block_2:
        ld      r4, [c]             ; c <- r4
        mul     r8, r4, r4          ; r8 <- r4 * r4
        st      r8, [d]             ; d <- r8
        jmp     block_5
block_3:
        st      r5, [b]             ; r5 <- b
        st      r5, [c]             ; c <- r5
        jmp     block_4
block_4:
        ld      r6, [c]             ; r6 <- c
        ld      r7, [c]             ; r7 <- c
        mul     r8, r6, r7          ; r8 <- r6 * r7
        st      r8, [d]             ; d <- r8
```

【図１５】


３１
操作者
３２
グラフィカルユーザ
インターフェース
ソースプログラム
記憶部
１０
４ コンパイル処理装置
指示情報受付部
１８
ソースプログラム
解析部
１１
１９
コンパイル
指示情報
記憶部
１２
ソースプログラム
解析情報
記憶部
１５
プロファイル情報
記憶部
１３
実行形式生成部
１４
プロファイル情報
生成部
２０
実行形式プログラム
記憶部
[実行]

【図１６】


ソースプログラム
記憶部
101
コンパイル処理装置
102
実行形式プログラム
記憶部
103
[実行]
101
ソースプログラム
記憶部
プロファイル情報
記憶部
104
コンパイル処理装置
102
実行形式プログラム
記憶部
(最適化されたもの)
103

【書類名】　　要約書

【要約】

【課題】　処理時間を短縮し、迅速に実行形式プログラムを生成することができるコンパイル処理装置を提供する。

【解決手段】　ソースプログラム解析部１１は、ソースプログラムに対する種々の解析処理を行って、ソースプログラムの解析情報を生成し、これをソースプログラム解析情報記憶部１２に記憶する。ソースプログラム解析情報記憶部１２の解析情報に基づいて、実行形式生成部１３によりプロファイル情報生成用実行形式プログラムを生成し、これを実行することにより得られる情報から、プロファイル情報生成部１４によりプロファイル情報を生成する。実行形式生成部１３は、ソースプログラム解析情報記憶部１２に記憶されているソースプログラムの解析情報と、プロファイル情報生成部１４により生成されたプロファイル情報とに基づいて、最適化した実行形式プログラムを生成する。

【選択図】　　図１

認定・付加情報

特許出願の番号	平成１１年　特許願　第１８８６６１号
受付番号	５９９００６３７１２６
書類名	特許願
担当官	第七担当上席　　００９６
作成日	平成１１年　７月　８日

＜認定情報・付加情報＞

【特許出願人】

【識別番号】	000002185
【住所又は居所】	東京都品川区北品川６丁目７番３５号
【氏名又は名称】	ソニー株式会社

【代理人】　申請人

【識別番号】	100091546
【住所又は居所】	東京都新宿区西新宿８丁目１２番１号　篠ビル８階　佐藤正美特許事務所
【氏名又は名称】	佐藤　正美

次頁無

出証特平１１－３０５３２７４

出願人履歴情報

識別番号　〔０００００２１８５〕

１．変更年月日　１９９０年　８月３０日
［変更理由］　新規登録
住　所　東京都品川区北品川６丁目７番３５号
氏　名　ソニー株式会社